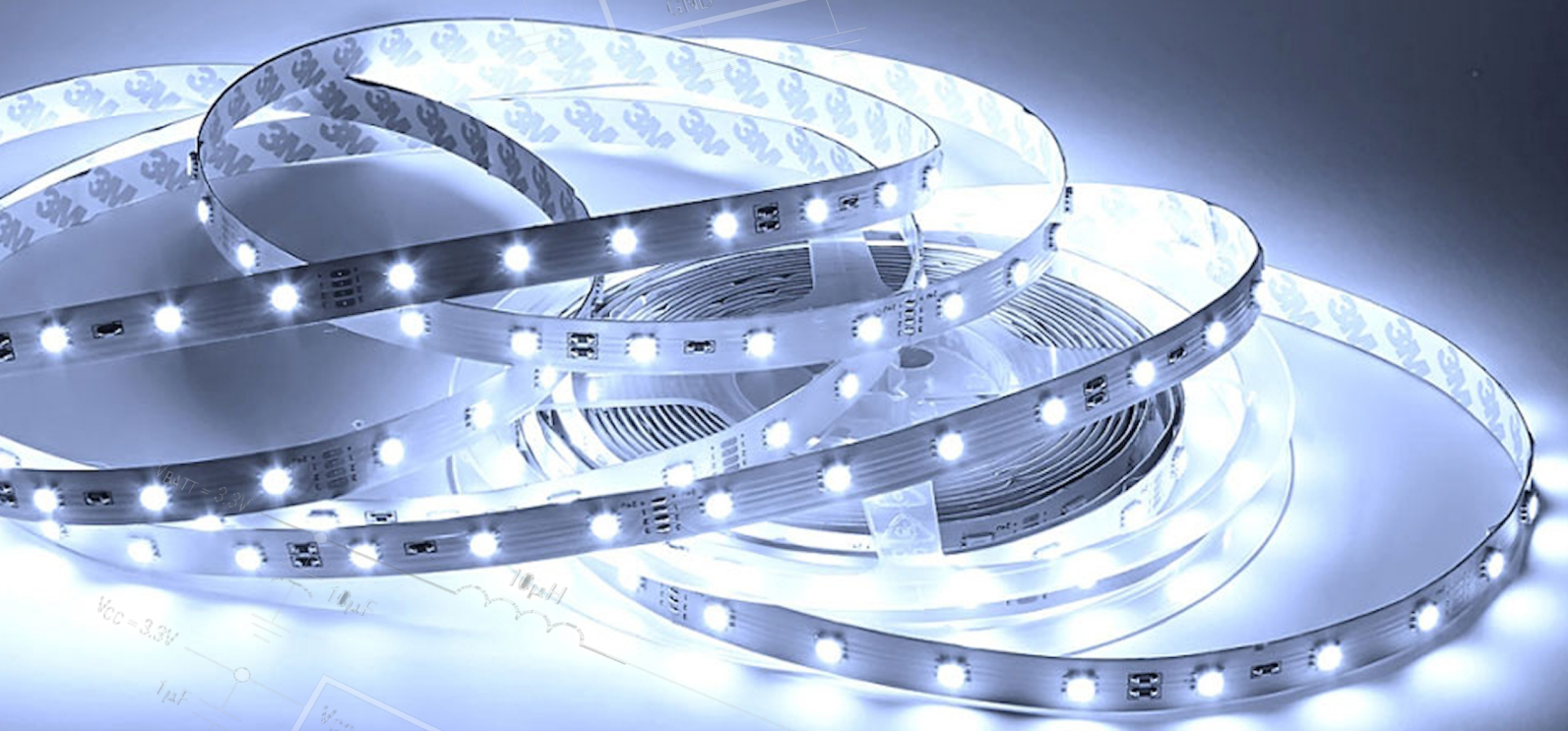
leggi Fare Elettronica su
IE cloud
www.ie-cloud.it

fare elettronica
n.346 - Giugno 2014

LED bianchi

come pilotarli a corrente costante



microcontrollori

Routine di Encryption

per le famiglie PIC24

I segreti dei PCB
Caricabatteria USB
Alimentatori per cellulari

Fare Elettronica n.346 – Giugno 2014

Sommario

Caricabatteria USB con ATmega *pag. 4*
Una connessione USB, un microcontrollore ATMega ed un convertitore DC/DC sono tutto quello che serve per realizzare un caricabatterie per celle a ioni di litio. Scopriamo come.

Sistema di monitoraggio batterie (IV parte) *pag. 11*
Concludiamo il nostro discorso presentando l'unità SLAVE, essa funzionerà in abbinamento alla scheda MASTER descritta nella puntata della volta scorsa. Entrambe le realizzazioni, fra loro dialoganti, costituiranno il nostro sistema di batteries monitor.

Routine di encryption per PIC24 *pag. 33*
Esistono tre standards di cifratura approvati per l'uso secondo il Federal Information Processing Standards (FIPS). Nell'articolo si descrive l'implementazione di due di questi per i PIC24 e i dsPIC30/33: il Triple Data Encryption Standard (TDES) e l' Advanced Encryption Standard (AES).

Fare circuiti stampati *pag. 48*
Le regole progettuali da tenere in considerazione per agevolare la fase di produzione dei circuiti stampati e le possibili scelte, in termini di produttori, sul territorio nazionale e non.

Alimentatori per Cellulari e dispositivi mobili *Pag.55*
Tanto diffusi da sembrare invisibili, possono costituire sorgente di utili componenti e svelare problematiche insospettate.

Pilotare i LED bianchi *Pag.61*
In questo articolo si parlerà di LED bianchi, del perché pilotarli a corrente costante e si metteranno a confronto sei diversi LED di due marchi. L'obbiettivo e capire pregi e difetti dei circuiti di pilotaggio dei LED.

Introduzione alle porte logiche di base *Pag.67*
E' ben noto a tutti che un sistema complesso digitale è costituito da una combinazione di porte logiche tra esse collegate, dette reti, che rispettano una ben determinata legge che esprime come risultato in uscita una combinazione o parte di essa dei segnali di ingresso. Vediamo come funzionano.

NEWS *pag. 78*
Microchip rilascia l' RN4020 Bluetooth Smart Module
Sintetizzatore frazionale N da 6GHz
TTI distribuisce i connettori Molex
NEC LCD Technology nomina Rutronik distributore dell'anno per il 2013

l'elettronica è qui.

Il nuovo spazio dedicato ai progettisti elettronici e ai makers

IE cloud
www.ie-cloud.it

INWARE EDIZIONI

Il nuovo portale IEcloud mette a disposizione degli utenti numerosi ed interessanti contenuti in tema di elettronica.
Progetti, articoli e news possono essere condivisi nella community e fruiti in tempo reale da tutti i membri.

IEcloud è il portale di riferimento per tutti i professionisti, progettisti, studenti e appassionati di elettronica.

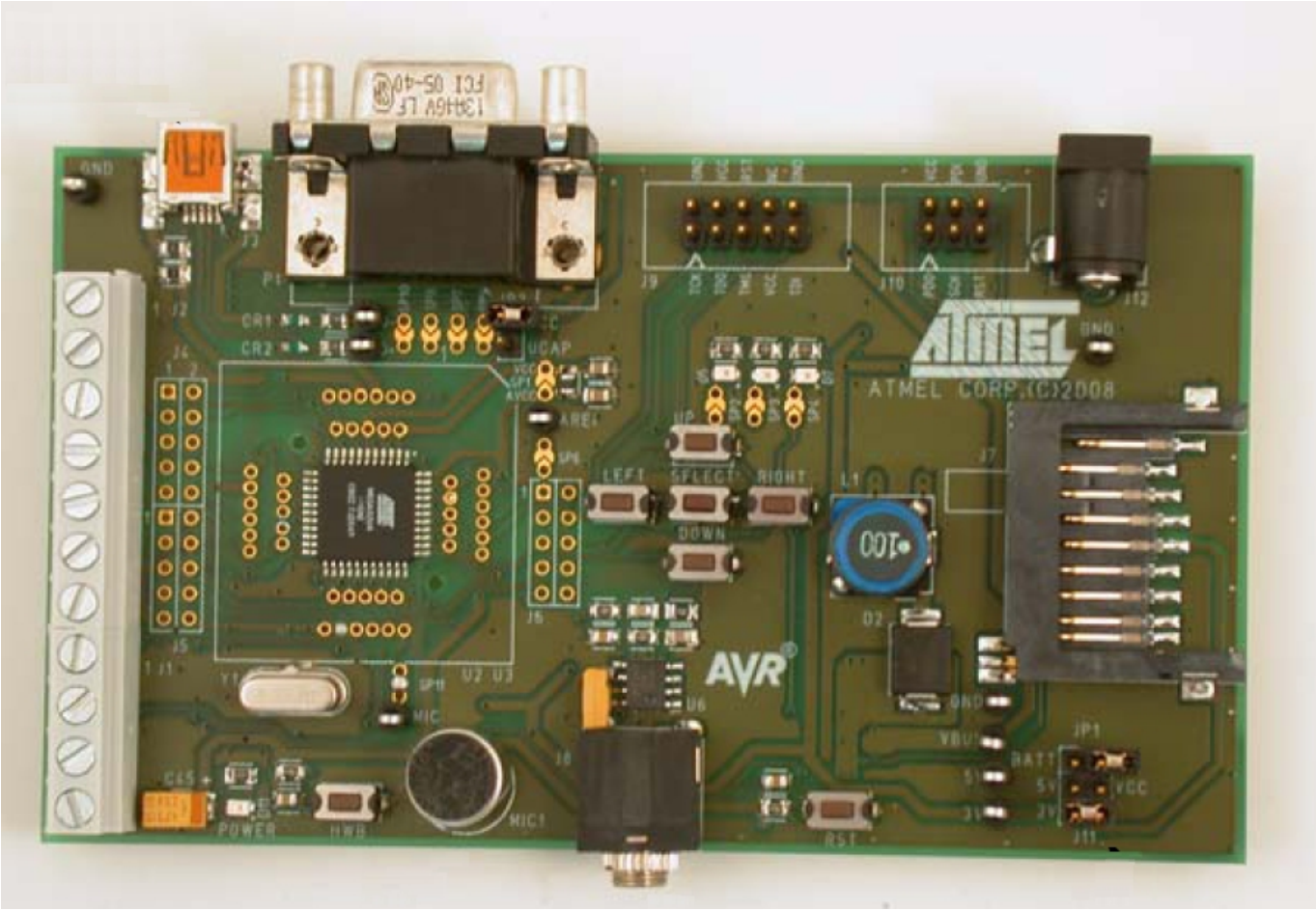
Registrati subito, è GRATIS!

- Centinaia di articoli, riviste, ebook, video, pdf sempre a tua disposizione
- Una community per condividere i propri progetti o per cercare collaborazioni
- Notizie, aggiornamenti ed eventi relativi al mondo dell'elettronica
- Un portale fruibile da qualsiasi dispositivo smartphone, tablet o PC

	FREE	SMART	MAKER	GENIUS
	GRATIS per sempre	€5.99/mese o 59,99/anno	€5.99/mese o 59,99/anno	€7.99/mese o 79,99/anno
Accesso a news ed eventi	✓	✓	✓	✓
Accesso alla community	✓	✓	✓	✓
Accesso ai progetti gratuiti	✓	✓	✓	✓
Accesso ai progetti premium	✗	✓	✓	✓
Accesso a tutte le riviste Fare Elettronica	✗	✗	✓	✓
Accesso a tutte le riviste Firmware	✗	✓	✗	✓
	Registrati	Acquista	Acquista	Acquista

Caricabatteria USB con ATmega

By Redazione | Fare Elettronica 346 – Giugno 2014



Una connessione USB, un microcontrollore ATMega ed un convertitore DC/DC sono tutto quello che serve per realizzare un caricabatterie per celle a ioni di litio. Scopriamo come.

Nel 2008 Atmel ha introdotto la nuova famiglia di microcontrollori ATmega16/32U4 che combinano interfaccia USB 2.0 e funzionalità analogiche ad elevate prestazioni. Grazie a queste caratteristiche, tali microcontrollori rappresentano una soluzione ideale per sistemi a basso costo e con alimentazione da porta USB per applicazioni come accessori per console di gioco (ad esempio gamepad) o caricabatterie. E proprio di ciò, un caricabatterie per celle a ioni di litio, parleremo in questo articolo. Di seguito sono riportati i punti salienti dell'applicazione che è basata sulla scheda di valutazione EVK527 per ATmega16/32U4; i dettagli sono discussi nell'Application Note di Atmel indicata nei riferimenti [1].

Il problema della carica delle batterie

Prima di presentare i dettagli dell'applicazione, può essere utile introdurre brevemente il problema della carica delle batterie a ioni di litio.

Le batterie agli ioni di litio sono la tecnologia oggi più diffusa per le celle ricaricabili. Tra i vantaggi principali vi sono una bassa resistenza interna, un tempo di vita elevato, tempi di carica inferiori,

l'assenza di un effetto memoria della cella, una più bassa auto-scarica, minore tossicità. Lo svantaggio principale è invece legato alla scarsa tolleranza dei dispositivi a sovra-carica; in questo caso potrebbero verificarsi elevati assorbimenti di corrente ed un rapido incremento della temperatura interna del dispositivo che potrebbe andare in fuga termica. Non sono rari i casi segnalati di batterie ricaricabili utilizzate in apparecchiature consumer (computer portatili, ad esempio) che sono esplose, provocando non pochi danni, proprio per problemi al circuito di carica.

La procedura di carica di una batteria agli ioni di litio prevede tre diverse fasi, come mostrato schematicamente in figura 1. Le prime due sono a corrente costante mentre la terza a tensione costante. Il passaggio tra le prima due fasi è dettato dal raggiungimento di soglie per la tensione della cella. Il processo di carica termina invece quando, nell'ultima fase di carica a tensione costante, la corrente assorbita, che decresce nel tempo, scende al di sotto di una soglia. I valori delle soglie e delle correnti di carica da applicare per le diverse fasi sono definiti dal costruttore e specifici della particolare serie di batterie; i valori in figura 1 sono da riferirsi ad una batteria della seria VARTA™ EasyPack da 550mAh.

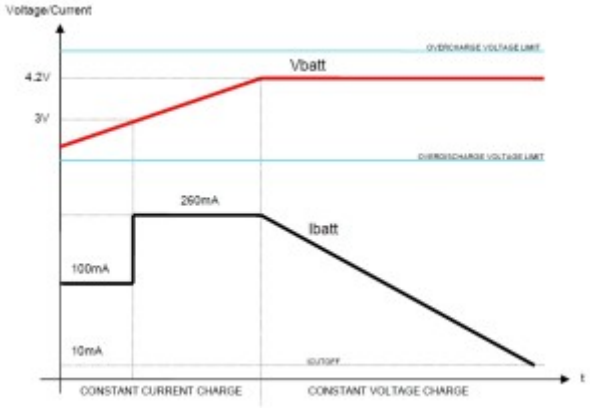


Figura 1 La procedura di carica di una celle a ioni di litio (da [1])

Il cuore del circuito : l'ATmega

Per realizzare il nostro circuito caricabatteria, tutto quello che ci occorre sono quindi una tensione di alimentazione in ingresso (ad esempio a 5V), un circuito di conversione DC/DC per generare la tensione di carica della batteria ed un microcontrollore per controllare il processo di carica secondo lo schema indicato in precedenza. Tutte queste risorse sono disponibili sulla scheda di sviluppo EVK527 mostrata in figura 2. La scheda, infatti, si basa su un micro ATmega32U4 con 32 KB di memoria Flash, porta USB 2.0 e ADC a 12 canali e dispone di un circuito di conversione DC/DC con topologia buck e controllo di tipo PWM generato direttamente dal microcontrollore. La porta USB può essere usata sia per derivare l'alimentazione per la scheda ed il circuito di carica della batteria che come linea di comunicazione con un terminale remoto.

L'ATmega32U4 è un microcontrollore con architettura RISC con capacità di calcolo fino a 16 MIPS; include un register file di 32 x 8 locazioni ed un moltiplicatore embedded in grado di eseguire un'operazione in 2 cicli di clock. Include 32KByte di memoria flash in-system self-programmable, 2.5 KByte di memoria SRAM ed EEPROM di dimensioni 1 KByte. Dispone, come detto, di una porta USB 2.0 in grado di sostenere data rate in trasferimento fino a 12 Mbps ed di un convertitore ADC a 12 canali e 10-bit di risoluzione. Integra un circuito PLL, timer e counter ad 8 e 16 bit, diversi canali PWM anche per applicazioni high-speed, interfaccia SPI master/slave e porta USART programmabile con controllo di flusso hardware. Supporta tensioni di alimentazione tra 2.7 V e 5.5V; la corrente assorbita, ad una frequenza di clock di 4 MHz, è di solo 2.2 mA, rendendo così ideale il dispositivo per applicazioni con alimentazione da porta USB.

L'Application Note [3] mostra come implementare una porta di comunicazione virtuale CDC (communication device class) mediante l'interfaccia USB, realizzando così un semplice convertitore USB-seriale tramite l'ATmega. Questa stessa funzionalità è usata nel caso del nostro caricabatteria per stabilire un canale di comunicazione tra la scheda di sviluppo EVK527 ed un terminale remoto per l'invio di informazioni relative allo stato del processo di carica. Il convertitore ADC dell'ATmega è invece usato per controllare costantemente lo stato di carica della batteria.

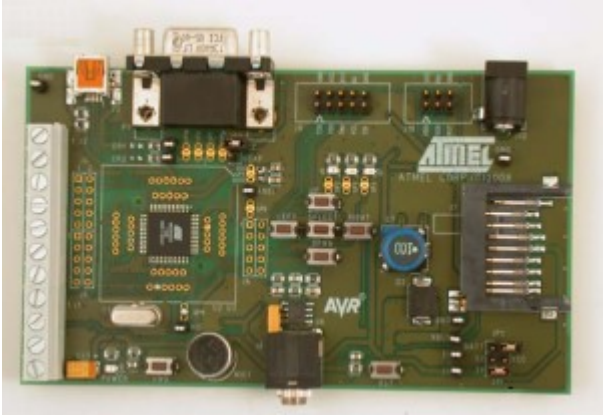


Figura 2 La scheda di sviluppo EVK527 per ATmega32U4

Il circuito di conversione DC/DC della tensione di alimentazione

Come accennato in precedenza, per generare la tensione di carica della batteria occorre un circuito di conversione della tensione DC/DC. La scheda utilizza un convertitore buck il cui schema di principio è mostrato in figura 3.

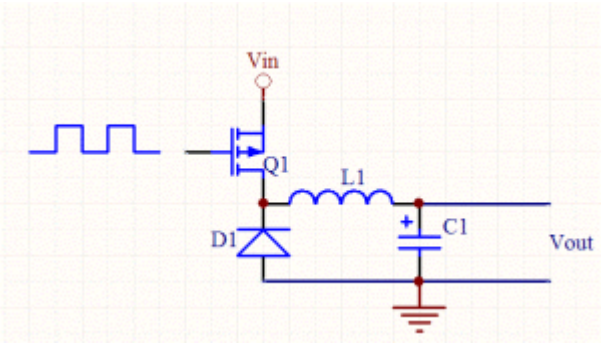


Figura 3 Schema di principio di un convertitore buck

I convertitori buck sono convertitori in continua di tipo step-down caratterizzati da elevata efficienza (tipicamente maggiore del 95%). Il principio di funzionamento su cui si basano è piuttosto semplice; un induttore viene caricato e scaricato alternativamente attraverso un diodo ed uno switch (tipicamente un transistor) per trasferire energia dall'ingresso verso il carico. Se la corrente attraverso l'induttore non si riduce mai a zero, il convertitore opera in modalità continua seconda lo schema temporale riportato in figura 4a. In questo caso, quando lo switch è chiuso il diodo è contro polarizzato; quindi non conduce. L'induttore assorbe energia dall'ingresso e la corrente aumenta linearmente. Quanto lo switch viene aperto, il diodo viene forzato in conduzione e l'induttore eroga l'energia immagazzinata; l'intensità della corrente decresce linearmente nel tempo. Il rapporto tra la tensione di ingresso e quella di uscita è dato dal duty cycle del periodo di commutazione dello switch. Se invece l'induttore si scarica completamente durante l'intervallo di off dello switch, il convertitore opera in modalità discontinua (figura 4b); la tensione di uscita in questo caso è funzione anche del valore dell'induttore, del periodo di commutazione dello switch e della corrente assorbita dal carico.

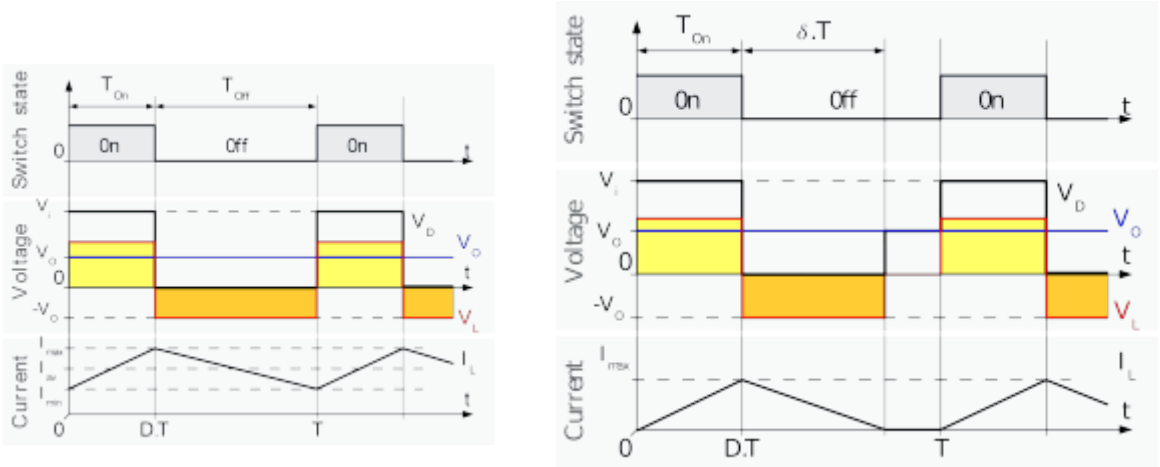


Figura 4 Principio di funzionamento di un convertitore buck nelle modalità continua (a) e discontinua (b) (da wikipedia)

Nel nostro caso, lo switch del convertitore buck viene controllato direttamente dal microcontrollore sfruttando una delle uscite PWM disponibili a bordo dell'ATmega; in questo modo sarà possibile controllare in maniera precisa il duty-cycle del periodo di commutazione e quindi, per quanto osservato in precedenza, la tensione di uscita del circuito. Il clock per il circuito PWM viene generato internamente al microcontrollore mediante PLL e post-scaler; la frequenza impostata è la massima consentita di 64 MHz, così da ottenere la massima risoluzione possibile nella modulazione del duty-cycle. Il segnale PWM di controllo risultante ha un periodo 250 KHz.

Il software applicativo

Il software applicativo del nostro carica-batteria può essere scaricato dall'indirizzo web indicato nei riferimenti [2]. Prima di usare la scheda di sviluppo per applicazioni proprie, tenere presente che l'esempio è stato sviluppato per una particolare batteria Varta; nel caso quindi di batterie diverse andranno modificati opportuni parametri del software onde evitare di danneggiare il nostro dispositivo. Nell'Application Note [1] ed all'interno dello stesso codice sorgente [2] sono chiaramente riportati i punti dove apportare le modifiche necessarie.

Il seguente listato 1 riporta la funzione *main()* dell'applicativo che è stato sviluppato per gestire il caricabatteria.

Listato 1

```
int main(void)
{
   Usb_enable_regulator();
   wdtdrv_disable();
   start_boot_if_required();
```

```
    Clear_prescaler();

    scheduler();

    return 0;

}
```

Come si vede è piuttosto semplice e schematica. La funzione *Usb_enable_regulator()* abilita il regolatore interno per i pads USB; tipicamente viene disabilitato per applicazioni con tensioni di alimentazione inferiore a 3.5V per ottimizzare la dissipazione di potenza. Quindi, la funzione *wdtdrv_disable()* disabilita il watchdog timer del microcontrollore. In seguito viene avviato il boot-loader – *start_boot_if_required()* – e resettato il prescaler del clock interno della CPU – *clear_prescaler()*. Infine viene attivato lo *scheduler()*. Tale scheduler è un loop infinito che richiama in sequenza i task che sono stati abilitati all'interno del file *conf_scheduler.h*; non è implementato nessuno schema di pianificazione in tempo reale ma, semplicemente, al termine di un task è chiamato quello immediatamente successivo nella sequenza predefinita. Il ciclo infinito è preceduto dalla inizializzazione dei diversi task previsti mediante chiamata della apposite funzioni relative.

Nella nostra particolare applicazione sono previsti i seguenti tre task : *usb_task()*, *cdc_task()* e *batt_task()*. I primi due si riferiscono alla gestione della porta seriale e della comunicazione con il terminale remoto. Il terzo invece controlla il processo di carica della batteria. Il cuore della funzione è una macchina a stati implementata secondo il diagramma di flusso riportato in figura 5.

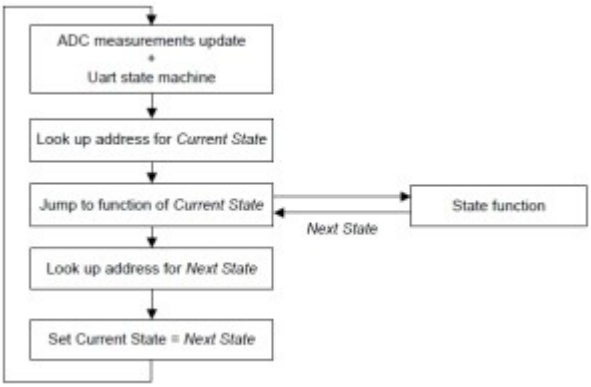


Figura 5 Diagramma di flusso della macchina a stati della funzione batt_task()

Il listato 2 evidenzia la parte di codice corrispondente all'interno del file sorgente *batt_task.c*

Listato 2

```
123. unsigned char CurrentState;     //!< \brief Global that indicates current state
124. //!<
125. //!< Updated by main().
126. //!< \note See menu.h for definition of states.
127. unsigned char nextstate, inp;
128. unsigned char (*pStateFunc)(unsigned char);   // Function pointer.
…

397. //wait a complete ADC conversion to update batt measurements
398. ADC_Wait();
399. // Run function associated with current state, get next state in return.
400. if (pStateFunc != NULL){
```

```
401. nextstate = pStateFunc(inp);
402. }
403. // Look up function for next state, if it differs from the current.
404. if (nextstate != CurrentState) {
405. CurrentState = nextstate;
406. for ( i = 0; menu_state[i].state != 0; i++) {
407. if (menu_state[i].state == CurrentState) {
408. pStateFunc = menu_state[i].pFunc;
409. }
410. }
411. }
```

Ad ogni stato, come si vede, è associato un puntatore a funzione che viene usato per chiamare la funzione appropriata all'inizio del ciclo. Ognuna delle funzioni ritorna a sua volta il prossimo stato. Il ciclo inizia con la chiamata alla funzione *ADC_Wait()* che consente di aggiornare i parametri dello stato di carica della batteria; questa stessa funzione viene anche chiamata ripetutamente all'interno delle altre funzioni di stato per verificare come proseguire nel processo. Nell'Application Note [2] sono chiaramente descritti tutti gli stati previsti e le funzioni associate. La funzione *charge()*, in particolare, gestisce il processo di carica nelle diverse fasi di pre-qualifica, carica a corrente costante e carica a tensione costante in accordo allo schema discusso inizialmente.

Riferimenti

[1] "AVR146: Lithium-Ion Battery Charging via USB with ATmega16/32U4"

[2] http://www.atmel.com/dyn/resources/prod_documents/EVK527-series4-cdc_batt_chg-2_0_1.zip

[3] http://www.atmel.com/dyn/resources/prod_documents/EVK527-series4-cdc-2_0_2-doc.zip

THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT

Stencil gratuito
con ogni ordinazione di prototipi PCB

MAGIC-PCB
RFID inside
NUOVO!

Embedded RFID
per convalidare, controllare e proteggere il tuo prodotto
www.magic-pcb.com

p-cad 2006 · TARGET 3001! · Altium Designer · EDWIN · cadence · GraphiCode · PROTEUS · NATIONAL INSTRUMENTS · KICAD · DIPTRACE · RS-274-X · Easy-PC · Sprint Layout · PULSONIX · BoardMaker3

www.pcb-pool.com

Beta LAYOUT
create : electronics

PCB-POOL® è un marchio registrato di Beta LAYOUT GmbH

eSTORE®
Beta LAYOUT

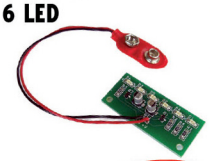
Realizzazione, saldatura, montaggio

Big Beta-Reflow-Kit

Video
www.beta-estore.com/video

Reflow-Controller

€ 129,00

Lampeggiatore 6 LED

€ 6,00

Arduino Mega (ATMega 1280-16AU) compatibel

€ 36,50

Toolkit Extended

€ 149,00

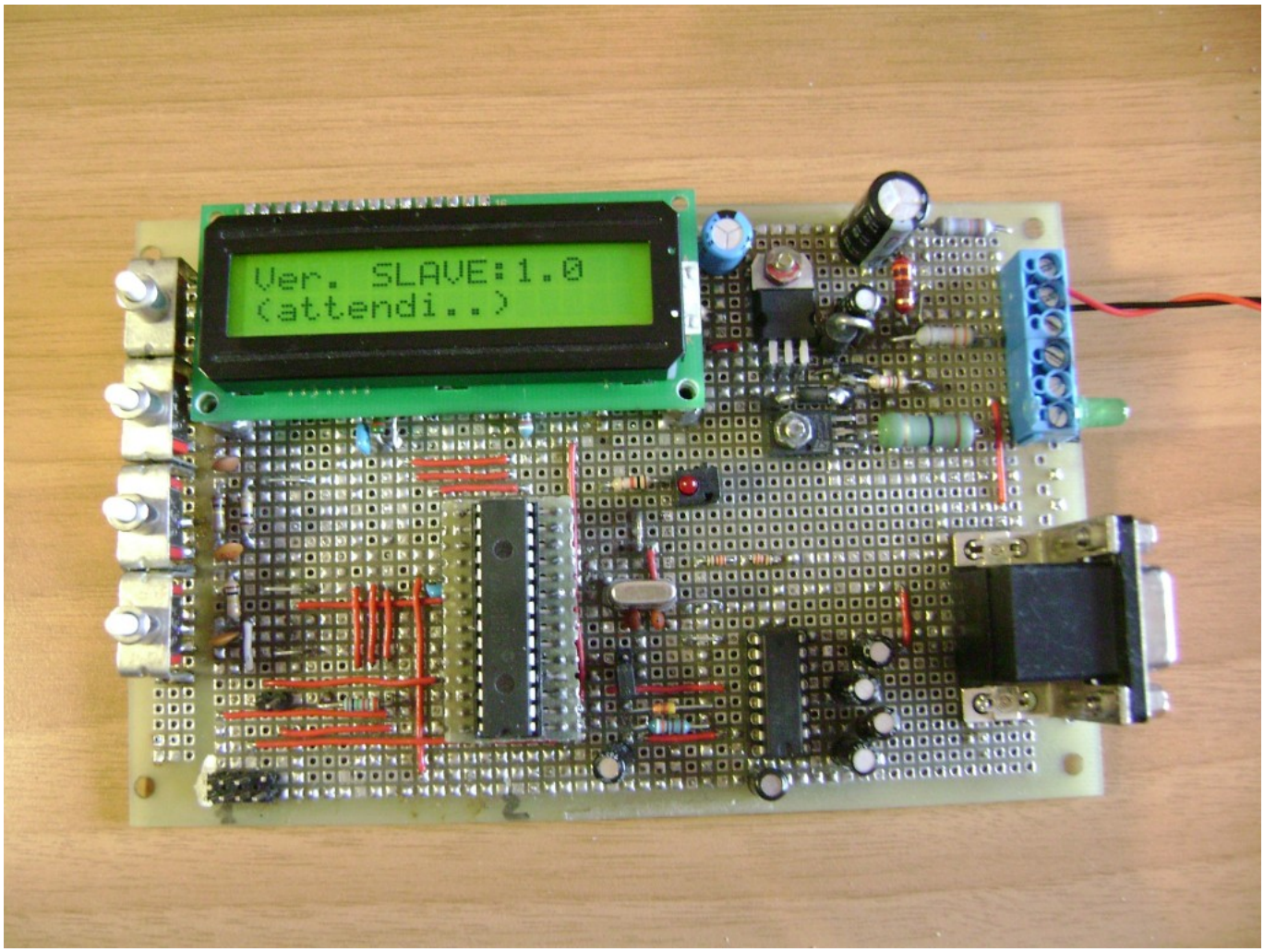
www.beta-eSTORE.com

Beta LAYOUT
create : electronics

eSTORE® è un marchio registrato di Beta LAYOUT GmbH

Sistema di monitoraggio batterie (IV parte)

By Grazia Ancona | Fare Elettronica 346 – Giugno 2014



Concludiamo il nostro discorso presentando l'unità SLAVE, essa funzionerà in abbinamento alla scheda MASTER descritta nella puntata della volta scorsa. Entrambe le realizzazioni, fra loro dialoganti, costituiranno il nostro sistema di batteries monitor.

Una panoramica delle funzioni della scheda SLAVE

Coloro che hanno seguito, nel corso della puntata scorsa, l'evoluzione della scheda MASTER ricorderanno che la funzione di quest'ultima era di raccogliere una serie di misure ed inviarle, in forma di dati seriali, ad un'altra unità: la SLAVE. Lo scopo dell'attuale puntata, conclusiva del nostro discorso, è proprio quello di presentare quest'ultima unità che, complementando le funzioni del MASTER, ha lo scopo di visualizzare i dati ricevuti e, nell'eventualità, gestire azioni di allarme e di avvertimento per l'utente. Vorrei soffermarmi su quest'ultima prerogativa poiché, sia detto per inciso, l'utente non può sorvegliare il display continuamente e, in caso di misure "sospette", deve comunque essere avvertito per poi intraprendere l'azione che egli riterrà opportuna. Il diagramma di flusso delle varie funzioni della scheda SLAVE è visibile in figura 1.

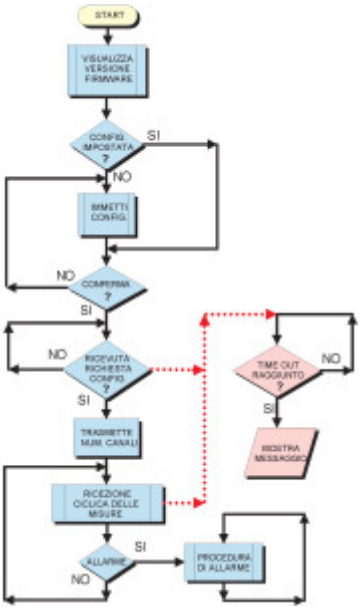


Figura 1: Diagramma di flusso semplificato delle funzioni della scheda SLAVE

Non essendo possibile, per ovvi motivi tipografici, illustrare il flow-chart completo, in figura 1 appare solo un diagramma semplificato dove l'andamento del flusso è riassunto nelle funzioni principali. L'interazione fra MASTER e SLAVE, servita da un apposito protocollo a livello di carattere ASCII (ne parleremo in dettaglio più avanti), è stata descritta nella scorsa puntata. Nel corso di quest'ultima, abbiamo anche presentato un diagramma a stati (vedi figura 13 della terza puntata) in cui le varie fasi del "dialogo" fra le due unità erano rese graficamente. Osservando il flow-chart di figura 1, appare chiaro che il compito principale della scheda SLAVE è quello di presentare le misure acquisite in una forma grafica facilmente verificabile dall'utente, calcolare la tensione totale della serie e gestire i vari livelli di allarme o di errore di comunicazione. Un'altra importante funzione, che non appare nel flow-chart poiché non fa parte del firmware, è quella di fornire l'alimentazione principale alla scheda MASTER. In particolare, si parlerà di tele-alimentazione poiché essa perviene a quest'ultima unità tramite un cavo dotato di 5 conduttori e, se possibile, di uno schermo dotato di terminazione filare. La presenza di quest'ultimo è opinabile ma, specie in ambienti montani o industriali, fortemente consigliata (vedi considerazioni fatte nella terza puntata). Del resto i normali cavi per impianti antifurto possono essere dotati di un rivestimento schermato che, nel nostro caso, dovrebbe essere collegato elettricamente alla terra della rete luce (vedi figura 3 della terza puntata). A proposito del cavo, per piccole distanze va bene un tipo a 5 conduttori da 0,22 mmq. Se la distanza aumenta (oltre i 12 m.), possiamo prevedere un cavo a 4 conduttori formato da 2 fili da 0,50 mmq per la tele-alimentazione dell'unità MASTER e 3 fili da 2 x 0,22 mmq per i dati (RS232) e per l'eventuale terra. Entrambe le tipologie di cavi sono facilmente reperibili in quanto, come accennato, sono usatissime dagli installatori. Tornando all'unità SLAVE, dobbiamo considerare un'altra importante funzione: la ricarica e il controllo dell'accumulatore di back-up. La presenza di quest'ultimo consente il funzionamento del nostro sistema di misura anche senza la tensione di rete e, cosa importante, senza attingere risorse energetiche dalla serie di batterie dell'impianto. La tensione di questo accumulatore è monitorata dal microcontrollore e, nel caso scendesse sotto un certo livello di guardia (circa 6,5 V), è prevista l'accensione di un LED d'allarme. L'unità SLAVE è dotata di 3 pulsanti che consentono, tramite semplici funzioni assistite dal firmware, l'impostazione dei dati da parte dell'utente. Le misure ed i vari messaggi sono visualizzati su un display LCD con matrice di 16 caratteri su 2 righe dotato di retroilluminazione. Si tratta di un componente molto economico, reperibilissimo, sicuramente già presente nel cassetto dell'auto-costruttore. In figura 2 possiamo trovare lo schema a blocchi della scheda SLAVE.

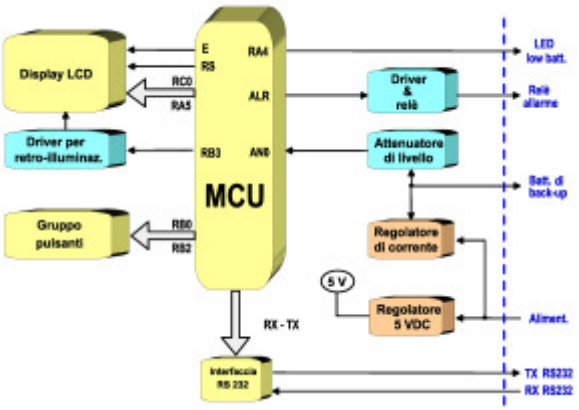


Figura 2: Schema a blocchi della scheda SLAVE

Descrizione dello schema elettrico dell'unità SLAVE

Osservando lo schema elettrico di figura 3, possiamo notare che il cuore del dispositivo è un microcontrollore PIC18F2550 con clock a 4 MHz e reset servito da una rete RC.

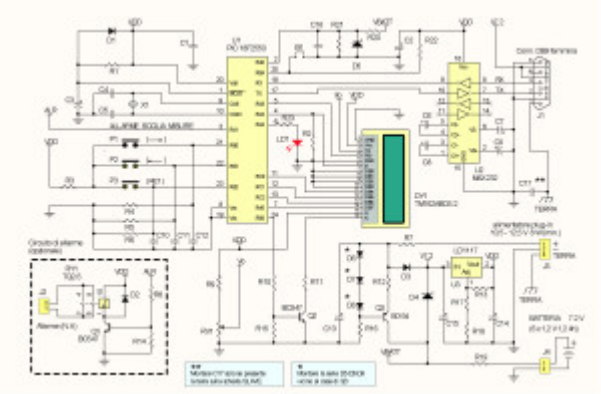


Figura 3: Schema elettrico della scheda SLAVE

La scelta di questo modello di MCU, avente 28 pin con case in dual-in-line, è stata condizionata da esigenze di memoria, disponibilità di pin di I/O ed economia d'acquisto. L'MCU accede al display mediante un BUS a 4 bit e fornendo i canonici segnali di controllo (E ed RS nello schema). Il bit di controllo R/W del display è posto a massa, ovvero il dispositivo rimane comunque e sempre nella modalità di scrittura dati. Questa condizione è posta dal momento che la MCU non accede alla lettura della RAM del display (per ulteriori chiarimenti, vedi l'ottimo articolo di A. Giannico sul numero 331 di FE). Il display, come accennato, dispone di una retroilluminazione a LED che viene controllata dall'MCU a mezzo di un semplice transistor bipolare (Q2 nello schema elettrico). In questo modo, la retroilluminazione sarà attiva solo nei casi elencati nella tabella 1.

Condizioni per l'accensione della retroilluminazione del display LCD		
Evento	Durata accensione retroilluminazione	Note
Avvio sistema	Circa 4 secondi	Vengono visualizzate sul display le versioni firmware di entrambe le unità

Pressione del tasto “RET” per più di 1 sec. da parte dell’utente (durante la scansione del ciclo di misure)	Fino al termine di un ciclo completo di misure successivo a quello in corso	In questo modo l’utente può leggere le misure fornite in ambienti scarsamente illuminati
Time-out raggiunto sulla mancata ricezione dei dati inviati dall’unità MASTER	Fino all’invio di dati corretti da parte del MASTER	Il display lampeggerà con cadenza approssimativa di 1 sec.
Errore di comunicazione	Fino all’invio di una stringa riconosciuta dall’unità SLAVE	
Allarme su unità MASTER	A tempo indefinito (il sistema deve essere riavviato manualmente)	
Allarme dovuto al superamento della soglia impostata dall’utente (valido per ogni canale di misura)	Fino a quando tutti i canali hanno riportato misure superiori alla soglia impostata	Durante l’accensione del display è contemporaneamente attivo il relè di allarme

Tabella 1

Va da sé che il lampeggio del display è provocato per attirare l’attenzione dell’utente, infatti esso è avviato solo in condizioni di anomalia e di allarme. Il trimmer RV1 serve per regolare il contrasto del display e va regolato, una volta per tutte, al momento del collaudo della scheda. Il controllo della pulsantiera, formata da 3 soli elementi NA, è ottenuto con il metodo del “polling”, ovvero il microcontrollore esamina ciclicamente i 3 pin (RB0, RB1 ed RB2) impostati come ingressi digitali che fanno capo ai 3 pulsanti. Ogni pulsante è dotato di una rete RC antirimbalzo e, in aggiunta, la procedura di acquisizione prevista nel firmware tende a eliminare le incertezze del contatto elettrico. In altri termini, il pulsante è ritenuto realmente premuto solo se la chiusura del contatto supera un certo tempo e se, alla fine, sarà rilasciato. Ciò nonostante, consiglio di usare elementi di qualità per la pulsantiera, in special modo se la stessa è posta su autoveicoli. Esauriamo il discorso “periferiche” accennando brevemente all’interfaccia seriale. Essa è servita dal “popolare” MAX232 (U2) che questa volta si collegherà al connettore a 9 poli femmina (J1) con ricezione e trasmissione invertite rispetto al MASTER. La ragione di ciò è semplice: le due unità, per poter dialogare fra loro, devono avere la ricezione dell’una sulla trasmissione dell’altra e viceversa. Anche se i connettori, per la presenza della tele-alimentazione e dell’eventuale terra, non seguono lo standard RS-232 si è ritenuto opportuno seguire un criterio DCE – DTE nell’assegnazione dei pin relativi al transito dati. In particolare, l’unità MASTER ha un connettore con le linee disposte come DCE (ovvero nello stesso modo di un terminale o PC) e lo SLAVE come DTE (ovvero come una periferica, un modem, ecc.). Attenzione quindi a non collegare direttamente le terminazioni ad un computer poiché, lo ribadisco, essendo il connettore fuori standard ed attraversato da linee di alimentazione potrebbe causare danni allo stesso PC (o terminale). Naturalmente il collegamento è possibile per le sole linee dati, in tal caso possiamo procedere come previsto in figura 4.

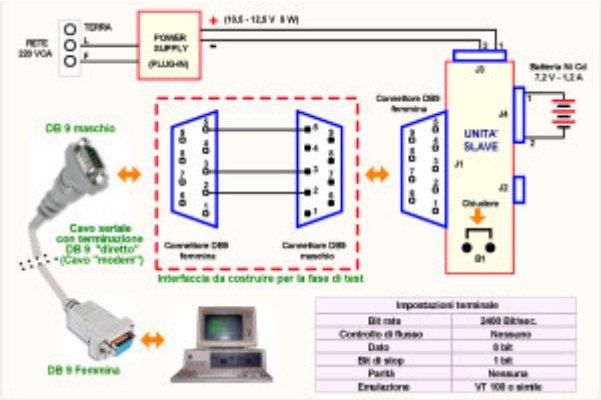


Figura 4: Connessioni per la fase di test

In questo caso, la connessione fisica avviene tramite una semplice interfaccia formata da due connettori DB-9 che hanno il compito di far giungere al terminale le sole linee dati con la massa. Il collegamento illustrato in figura 4 è comunque previsto per la fase di test della scheda SLAVE. Quest'ultima, come nel caso dell'unità MASTER, è guidata dal firmware e può essere attivata se all'avvio chiudiamo il ponticello B1. Durante il normale funzionamento, la chiusura o l'apertura di detto ponticello non causerà invece alcunché. Nella terza puntata, descrivendo l'unità MASTER, avevamo suggerito la possibilità di sostituire l'interfaccia RS-232 con un altro tipo più "performante" ed attuale. Queste considerazioni valgono anche nel caso della schede SLAVE, a patto che, mi scusino i lettori esperti per questa precisazione, entrambe le schede abbiano lo stesso tipo protocollo fisico. La batteria di back-up, composta da 6 elementi al Nichel-Cadmio da 1,2 Ah posti in serie, in condizioni di presenza dell'alimentazione esterna è ricaricata dallo stadio formato da Q3. Quest'ultimo, in abbinamento a pochi altri componenti, forma un generatore di corrente costante abbastanza stabile e preciso. La corrente di ricarica è mantenuta a circa 85 mA, valore che non consente, in caso di scarica completa, una ricarica rapida. Tuttavia questo aspetto non dovrebbe rappresentare un problema: i "fall-down" della rete luce non sono poi così prolungati quando avvengono. L'utente può intervenire modificando il valore della corrente di ricarica, nel caso voglia usare un accumulatore a capacità più elevata, rispettando la relazione:

Corrente di ricarica = 1,3 / R12

In cui il valore 1,3 è la tensione ai capi di R12. Questa è ottenuta sottraendo alla tensione ai capi dei 3 diodi (D6, D7 e D8) la Vbe di Q3. Essendo quest'ultima pari a circa 0,65 V e considerando una tensione diretta sui diodi di pari valore, avremo: ((0,65 V x 3) – 0,65 V) = 1,3 V. Sarà questa la tensione ai capi di R12, da cui applicando la legge di Ohm, è possibile ricavare la corrente circolante nel resistore ovvero quella immessa nella batteria. Si tenga presente, tuttavia, che la corrente di ricarica non deve superare 1/15 della corrente nominale dell'accumulatore. Questo significa che valori molto alti possono essere fuori dai parametri del progetto e, naturalmente, dalle correnti erogate dall'alimentatore plug-in. Moderazione quindi! Tornando ai diodi D6, D7 e D8, possiamo notare come siano polarizzati direttamente. Questo comporterà senz'altro una deriva termica negativa (con l'aumento della temperatura diminuisce la tensione diretta ai capi), un fattore che può essere sfruttato per rendere il circuito più stabile ricorrendo ad un semplice accorgimento. Sebbene il calore generato su Q3 sia abbastanza contenuto (è sufficiente la vite ed i dadi di fissaggio al PCB per smaltirlo) esso comunque avrà una temperatura tale da influenzare i componenti posti nelle sue vicinanze. Montando D6, D7 e D8 adiacenti al case di Q3, otterremo di contrastare la deriva termica di quest'ultimo per effetto dell'abbassamento della Vbe. Non è molto ma, come si dice dalle mie parti, "tutto fa brodo". La tensione di 5 V per l'alimentazione generale è invece stabilizzata da un regolatore a 3 terminali (U3) del tipo "low drop out", ovvero a bassa caduta di tensione interna. Esso raccoglie la tensione d'ingresso proveniente dall'alimentatore plug-in tramite il diodo D3, mentre la

tensione proveniente dalla batteria al Ni-Cd giunge al regolatore tramite il diodo Schottky D4. Questa configurazione è necessaria per evitare percorsi non voluti di corrente nel caso l'alimentazione da rete luce venisse meno. La presenza del diodo in tecnologia Schottky si rende invece necessaria in quanto, avendo la batteria un valore nominale pari a 7,2 V, siamo molto vicini al limite minimo della tensione d'ingresso del regolatore a 3 terminali. In questi casi, ovvero quando è la batteria a fornire l'energia, è giocoforza ridurre al minimo le cadute di tensione non volute. La tensione dell'accumulatore Ni-Cd è monitorata all'ingresso AN0 dell'MCU ad ogni misura di canale. Nel caso questa tensione fosse sotto una soglia di sicurezza (definita nella costante SOGLIA_LOW_BATT presente nel file header: master_standard.h) il LED LD1 comincerà a lampeggiare. Questo avvertirà l'utente che la batteria sta esaurendosi oppure che vi è un guasto nel circuito di ricarica. La tensione di batteria è misurata dall'MCU attraverso un partitore resistivo (formato da R20 ed R21) per adattarne il livello. Utilizzando una batteria con tensione nominale più elevata, sarà giocoforza rivedere i valori resistivi di questo partitore e modificare la costante definita RAPPORTO_DIV_MISURA (presente nel file header citato precedentemente) in base al nuovo rapporto di attenuazione. L'algoritmo di calcolo, del resto, è dettagliato negli esaurienti commenti. ll diodo ZENER D5 limiterà la tensione d'ingresso al microcontrollore nel caso la tensione di batteria dovesse superare i limiti progettuali, e lo possiamo considerare un'ulteriore protezione. Un cenno al circuito servito da Q1. Si tratta di un normale azionamento di relè ad uso allarme. Per ovvi motivi non può essere fatto direttamente dall'MCU e ciò richiede, per l'appunto, la presenza di un transistor bipolare. Vedremo meglio, quando parleremo del firmware, le funzioni di questo relè. Tuttavia il lettore sappia che, se non ha bisogno di ripetere il segnale d'allarme a distanza o azionare robusti avvisatori, questo circuito può essere omesso a beneficio della semplicità.

Realizzazione della scheda SLAVE

La costruzione della scheda SLAVE è più facile rispetto a quella dell'unità MASTER. Non vi sono particolari precauzioni sui percorsi di massa e l'unica difficoltà è relativa al montaggio del display e dei pulsanti. Questi ultimi, com'è evidente, dovranno essere accessibili da parte dell'utente. La parola "accessibile" dovrà essere ancora più restrittiva se il tutto sarà montato in zone scarsamente illuminate o ingombre di componenti impiantistici. Questo significa che:

- La parte accessibile del pulsante non deve avere meno di 1 cmq di area.
- I pulsanti devono essere fra loro distanziati di almeno 1 cm, meglio se di più.
- Le scritte relative alla funzione del pulsante devono essere sopra lo stesso (e non sotto) ed essere sufficientemente visibili.

La batteria di back-up dovrà essere posta sotto la scheda per evitare che eventuali trasudi di acidi possano intaccare le piste e/o i componenti. Si tenga presente di garantire una certa aerazione nel contenitore che ospiterà la scheda, sia Q3 che U3 possono infatti scaldare un pochino. Meglio sarebbe prevedere un piccolo dissipatore per quest'ultimo. Anche l'accumulatore al Ni-Cd potrebbe aumentare la temperatura interna, seppure in modo molto modesto in condizioni di funzionamento normale. Le morsettiere ed i connettori DB–9 possono essere montati sul retro del contenitore. La figura 5 ci darà un'idea di una possibile sistemazione delle parti mentre, la figura 6, mostra il prototipo sul quale sono state fatte le prove. Osservando il microcontrollore possiamo notare la presenza di un interfaccia costruito su vetronite forata, esso si rese necessario poiché, in "corso d'opera", fu cambiato il tipo di MCU. Tale scelta, imposta dalla disponibilità di memoria FLASH, impose di rivedere alcuni collegamenti. La presenza di 3 pulsanti non tragga in inganno: uno di loro è inutilizzato!

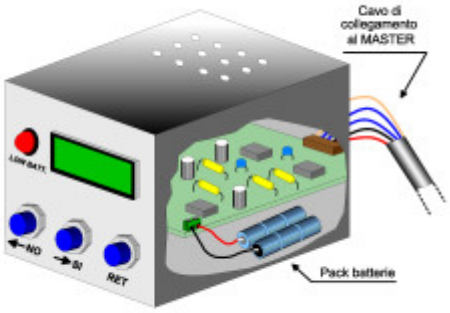


Figura 5: Possibile disposizione delle varie parti all'interno di un contenitore

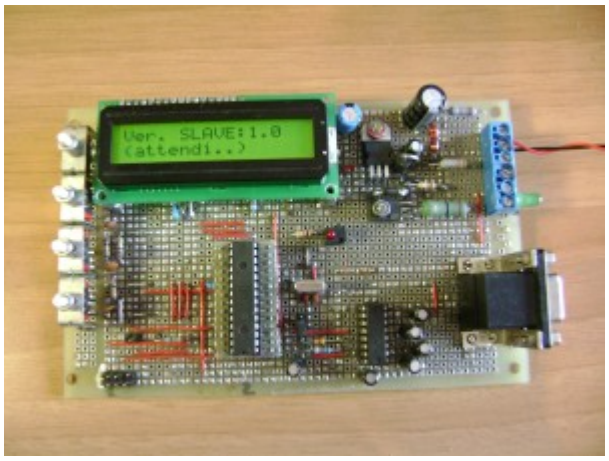


Figura 6: Prototipo montato su vetronite forata

L'ambiente di sviluppo

Prima di esaminare più da vicino le caratteristiche del programma, faremo una rapida "zoomata" sull'ambiente di sviluppo usato per la generazione del firmware. Ribadisco che non è lo scopo di questa serie di articoli entrare nel dettaglio di come possono essere organizzate e gestite le risorse di programmazione. Coloro che vorranno veramente approfondire l'argomento avranno l'imbarazzo della scelta fra le tante pubblicazioni dedicate a questa materia. Come accennato, per usare il sistema di misura così com'è, è sufficiente immettere nel microcontrollore il file .HEX fornito (nel nostro caso Slave.hex) usando uno dei tanti programmatori esistenti. Per coloro che hanno scelto questa strada, la lettura di questo paragrafo può terminare qui. Per i "coraggiosi" che invece vorranno modificare il codice sorgente (migliorare è quasi sempre possibile, adattare invece non ha il "quasi"…) daremo di seguito qualche preziosa indicazione. Anche il firmware dell'unità MASTER nasce da un codice sorgente elaborato in linguaggio C nel "dialetto" CCS. Le funzioni non-standard fornite da questo ambiente di programmazione sono molte e per intervenire sul codice sorgente è necessario disporre di un compilatore fornito, per l'appunto, dalla nota ditta CCS (Custom Computer Services). Ripetendo le stesse considerazioni fatte nella puntata precedente, a cui rimando comunque il lettore interessato, il compilatore dovrà lavorare in un adeguato ambiente di sviluppo. Noi abbiamo usato il PIC–C compiler per PIC18xx versione 4 e come In-Circuit-Debugger, (meglio conosciuto come ICD) l'ottimo PICKIT 3. E' quest'ultimo un prodotto che ha un altissimo rapporto prestazioni/prezzo e consente l'emulazione in real-time del programma comprendendo, tra l'altro, anche interessanti ed utili funzionalità di debug. Il collegamento della nostra applicazione, ovvero l'unità SLAVE, al PICKIT 3 dovrà avvenire secondo quanto illustrato in figura 7.

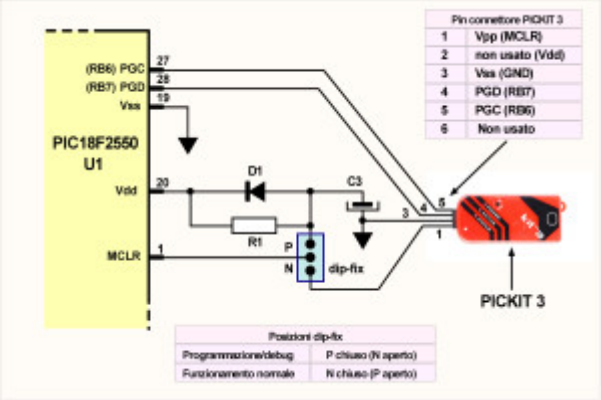


Figura 7: Collegamenti per la programmazione "in circuit"

Osservando lo schema di figura 7, si noterà la presenza del dip-fix a 3 pin che presenta 2 opzioni di chiusura, rispettivamente P ed N. Questa commutazione manuale, non riportata nello schema elettrico di figura 3, è necessaria dal momento che il pin di RESET dell'MCU ha dei componenti collegati (D1, R1e C3). Durante la fase di programmazione/debug, la linea di RESET (pin 1 del microcontrollore) deve invece ricevere, dal PICKIT 3, esclusivamente la tensione di programmazione (Vpp). Quest'ultima, ovviamente, non deve "vedere" componenti ma soltanto il pin 1 dell'MCU. Da qui la necessità di operare una commutazione manuale passando dalla programmazione/debug al funzionamento vero e proprio dell'unità. Infatti, nella prima eventualità, sarà lo stesso PICKIT 3 a fornire il necessario impulso di RESET. Durante il normale funzionamento della scheda, il RESET sarà invece fornito da gruppo RC (comprensivo di diodo). Come ribadito nella puntata precedente, non è lo scopo di queste pagine entrare nel dettaglio di come possono essere organizzate e gestite le risorse di programmazione. Per una più esaustiva visione dell'insieme si veda, a tal proposito, la figura 10 della terza puntata. Tuttavia mi pare il caso di spendere qualche parola per il lettore volenteroso. Il PICKIT 3 deve funzionare in un ambiente MICROCHIP, nel nostro caso l'ambiente di sviluppo scelto era L'MPLAB versione 8.xx. E' questo un applicativo, fornito gratuitamente dalla MICROCHIP, che consente, a fronte di qualche semplice impostazione, l'integrazione con compilatori in linguaggio C di case diverse. Fra le non tantissime opzioni previste, vi è per l'appunto quella relativa ai prodotti CCS. Intendiamoci, la terna MPLAB, compilatore CCS e PICKT 3 non è la sola possibile per ottenere un buon ambiente di sviluppo, tantomeno la più economica. E' solo fra le tante strade possibili, quella da noi percorsa. Del resto, la stessa CCS mette a disposizione un ottimo ICD che può funzionare nel suo ambiente di programmazione/debug. Per non parlare poi di prodotti di altre marche, facilmente rintracciabili nel commercio on-line e anche fra le pagine della nostra rivista.

Il firmware

Il programma presente nell'MCU deve poter svolgere le seguenti funzioni principali:

a) Ricevere le stringhe dati provenienti dall'unità MASTER e verificarne l'esattezza.

b) Presentare i dati sul display LCD mostrando, al contempo, le variazioni rispetto alle precedenti misure.

c) Calcolare e visualizzare la tensione totale della serie di batterie (dopo aver acquisito almeno un ciclo completo di misure).

d) Acquisire i dati dell'impianto immessi dall'utente verificandone l'esattezza.

e) Gestire gli allarmi per superamento delle soglie minime sui livelli di tensione delle batterie

dell'impianto.

f) Gestire gli allarmi catastrofici provenienti dall'unità MASTER.

g) Verificare il livello della batteria di back-up e, se troppo basso, avvertire l'utente con l'accensione di un LED.

Come possiamo notare, si tratta di un discreto impegno delle risorse firmware e questo ha imposto, nella scelta del microcontrollore, un elemento con una buona capacità di memoria FLASH-programma. Nella nostra esplicazione, partiremo dal punto a). Abbiamo accennato, quando abbiamo parlato del MASTER, come il dialogo fra le due unità presupponga l'implementazione di un semplice protocollo. Questi, nel nostro caso, sarà a livello di carattere in modo da poter utilizzare direttamente (e facilmente…) le risorse che le funzioni C mettono a disposizione del programmatore. Ricordiamo che un protocollo definito "a livello di carattere" consente esclusivamente la trasmissione-ricezione di caratteri ASCII a 7 o 8 bit (nel nostro caso a 8) con un bit di start ed uno (o due) di stop aggiunti alla stringa. Eventuali caratteri non stampabili (ad esempio il tasto ENTER, ecc.) saranno considerati dal terminale ricevente come caratteri di controllo ed interverranno sulla formattazione del testo o su altre funzionalità. Per avere un'idea dello scambio dati fra MASTER e SLAVE, sarà opportuno osservare il diagramma a bolle di figura 13 della terza puntata al quale rimando il lettore. Da esso si evince che le fasi di scambio dati sono 4 più una fase di allarme che, nel funzionamento normale, non dovrebbe avvenire. La tabella 2 ci mostrerà la direzione del flusso dei dati che, pur essendo bidirezionale, non può essere definito FULL DUPLEX in quanto la ricezione e la trasmissione non avvengono mai contemporaneamente.

Modalità di scambio dati fra unità MASTER e SLAVE			
MASTER	**DIREZIONE DATI**	**SLAVE**	**NOTE**
Trasmette la versione del proprio firmware	→	Riceve la versione e la visualizza	Vengono visualizzate le versioni firmware del MASTER e dello SLAVE per un tempo di 2 sec.
Trasmette la richiesta del numero di canali impostato dall'utente (configurazione)	→	Riceve la richiesta di configurazione	Il MASTER trasmette la richiesta a tempo indefinito (in questo tempo l'utente può immettere la configurazione)
Riceve la configurazione impostata	←	Trasmette il numero di canali (configurazione) non appena riceve la richiesta	
Trasmette dati di misura per ogni canale	→	Riceve le misure sui vari canali e le visualizza	Ciclo infinito che può essere interrotto solo con lo spegnimento del sistema o con un allarme catastrofico
Trasmette identificatore di allarme catastrofico	→	Riceve identificatore di allarme catastrofico e visualizza la condizione	La trasmissione dell'identificatore di allarme può avvenire in ogni stringa ed in ogni fase dello scambio dati

Tabella 2

MASTER e SLAVE, divenuti rispettivamente DCE e DTE dal punto di vista del collegamento seriale, eseguono dei controlli sulle stringhe ricevute in modo da escludere eventuali errori di trasmissione. Ovviamente, una stringa contenente un errore di trasmissione è considerata non valida ed i suoi dati saranno scartati. Il protocollo elaborato, in virtù della sua semplicità, prevede, nel caso dello SLAVE-DTE, la trasmissione di un carattere identificatore seguito da un campo di dati, anch'essi trasmessi

sotto forma di caratteri. L'unica forma di dato trasmesso dal MASTER-DCE è invece un singolo carattere (senza elemento identificatore iniziale), esso esprimerà un numero da 0 a 15 in esadecimale che rappresenta in numero degli ingressi di misura. La tabella 3 A-B chiarirà quanto esposto.

Dati trasmessi dall'unità SLAVE (DTE)							
Carattere Identificatore	**Campo dei dati**						**Descrizione contenuto del campo dati**
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
'V'	Cifra da '0' a '9'	'.'	Cifra da '0' a '9'	Carattere NULL	-	-	Versione (prima cifra) ed edizione (seconda cifra, separata da un punto) del firmware contenuto nell'unità MASTER
'R'	Carattere NULL	-	-	-	-	-	Richiesta della configurazione (numero di canali impostato dall'utente)
'M'	Cifra da '0' a '9'	Cifra da '0' a '9'	'.'	Cifra da '0' a '9'	Cifra esadecimale da '0' a 'F'	Carattere NULL	Valore in volt della misura su un ingresso (con una sola cifra decimale separata da punto) e numero di canale a cui fa riferimento la misura (in cifra esadecimale da '0' a 'F')
'X'	Carattere NULL	-	-	-	-	-	Allarme catastrofico sull'unità MASTER

Tabella 3 A

Dati trasmessi dall'unità MASTER (DCE)		
Byte 1	Byte 2	**Descrizione stringa**
Cifra da '0' a 'F' in esadecimale	Carattere NULL	Numero di canali impostato dall'utente (configurazione) espresso in esadecimale da 0 a F

Tabella 3 B

Come esposto, la massima configurazione possibile è pari a 16 canali mentre, osservando la tabella 3, notiamo che il numero massimo di canali, espresso in esadecimale, è pari ad 'F', ovvero a 15. Questa apparente incongruenza, dovuta all'esigenza di trasmettere un solo byte (e non due, ovvero '1' e '0' in HEX nel caso di 16 canali di misura) è risolta dal firmware dell'unità MASTER che aggiungerà 1 alla configurazione in arrivo dallo SLAVE. Quindi, il nostro numero 'F' corrisponderà a 16 canali, il numero 0 ad un solo canale e così via. Osservando la tabella 3 A, noteremo come alcune messaggi non contengono nemmeno la parte dati poiché essa sarebbe superflua mentre ogni stringa è terminata dal carattere NULL (definito anche '\0') che non può essere visualizzato. La presenza di quest'ultimo, non strettamente necessaria, è richiesta per ragioni di compatibilità con le procedure di ricezione. La versione ed edizione del firmware, dato trasmesso allo SLAVE per essere visualizzato all'accensione del sistema, è prevista per tenere sotto controllo le eventuali modifiche fatte al programma originario. Ne consiglio vivamente l'aggiornamento in quest'ultimo caso.

Altrimenti può essere lasciata così com'è, ovvero il canonico '1.0' della prima emissione. Accennavamo al controllo d'integrità dei dati. Non essendo previsto, per evidenti motivi di semplicità, un byte dedicato al cheksum della stringa trasmessa, il tutto si ricondurrà ad una verifica di congruità dei dati inviati. Facciamo un esempio osservando di nuovo la tabella 3 A, in particolare nel caso della stringa contenente la misura. Dando per scontato che l'identificatore è riconosciuto, nel nostro esempio il carattere 'M', il programma passerà a controllare i bytes del campo dati uno ad uno. In particolare il secondo byte deve contenere la prima cifra del valore della misura (cifra delle decine). Noi però sappiamo che il range permesso a quest'ultima può spaziare da 0 a 17 V (vedi le caratteristiche dl sistema descritte nella terza puntata). In altri termini possiamo aspettarci qualunque valore compreso fra '00.0' e '17.0'. Quindi, se la prima cifra non sarà uno '0' o un '1' il programma dedurrà di essere in presenza di un errore di trasmissione. E lo stesso accadrà se la prima cifra sarà, ad esempio, un carattere alfabetico. Questo criterio di verifica è graficamente illustrato nel diagramma di flusso di figura 8.

Figura 8: Diagramma di flusso delle verifiche sui dati ricevuti

Una volta che tutti i bytes sono stati ricevuti correttamente, la stringa sarà resa disponibile in un apposito buffer per essere poi elaborata dal programma. Tuttavia, data la bassa velocità della trasmissione asincrona, errori di trasmissione sono veramente sporadici. In ogni caso, il display dello SLAVE avviserà l'utente di questa eventualità accendendo la sua retro-illuminazione e visualizzando un messaggio. Anche il venir meno della trasmissione del MASTER sarà segnalato, dopo un tempo congruo di circa 17 secondi, con una scritta sul display. Entrambe le segnalazioni spariranno alla prima stringa ricevuta priva di errori. Quest'ultima funzionalità, che noi chiameremo TIME-OUT, è ottenuta per mezzo di un processo "parallelo" al flusso del programma: l'attivazione di uno dei timer interni all'MCU. Se il dato non arriva entro un tempo definito, la temporizzazione avviata provocherà un'interruzione e l'utente sarà avvisato del TIME-OUT sulla ricezione. Veniamo adesso a come saranno presentati all'utente i dati di misura inviati dal MASTER. La visualizzazione delle misure sul display è organizzata in modo tale che l'utente, in qualsiasi momento, possa conoscere:

- La misura effettuata sul canale attuale.
- La variazione della misura rispetto alla misura precedente sullo stesso canale, nel nostro articolo questa funzionalità è definita "tendenza"
- L'avviso se il canale misurato è sotto la soglia di allarme per bassa tensione.
  - La tensione totale della serie batterie (aggiornata ogni fine ciclo di misure).

La tendenza della misura è espressa con un segno "-" se la misura attuale è inferiore alla precedente e con un segno "+" se invece è maggiore. Un segno "=" indica, ovviamente, che non vi è stata una variazione significativa. La lettera "A" indica invece che la misura attuale è sotto la soglia d'allarme impostata dall'utente (condizione che attiva il relè di allarme remoto). Ho usato non a caso l'aggettivo "significativo" in quanto l'apprezzamento delle variazioni di misura è soggetto ad un piccolo valore di soglia d'isteresi. Questa tecnica permette di evitare variazioni continue nella visualizzazione quando, come spesso accade, ci si trova in prossimità di un valore di soglia. Lo stesso principio è applicato nell'esame del superamento delle soglie di allarme che, come accennato, azionano anche il relè previsto per avvisi remoti. La figura 8 A-B dovrebbe aiutarci a comprendere come l'utilizzo dell'isteresi renda più stabile proprio quest'ultima funzione e, più in generale, qualunque misura soggetta ad una soglia d'intervento prefissata.

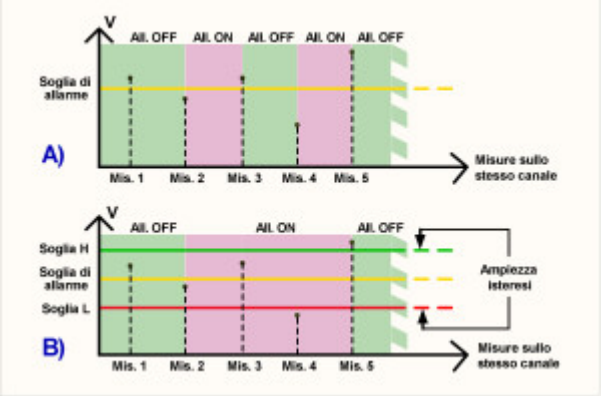


Figura 9 A-B: Applicazione di un valore d'isteresi nella gestione del relè di allarme

Osservando con attenzione la figura 9 A, possiamo notare come piccole variazioni della tensione misurata provochino continue commutazioni del relè. Nella figura 9 B, invece, il relè esegue due sole commutazioni poiché sono aggiunti due valori d'isteresi, uno superiore e l'altro inferiore, alla soglia d'intervento stabilita. Esaurita questa breve parentesi di chiarimento, prima di tornare con maggiore attenzione sull'argomento della visualizzazione delle misure, vediamo come possiamo configurare il nostro sistema. L'immissione dei dati dell'impianto è un operazione che l'utente dovrà necessariamente effettuare alla prima accensione del sistema. Quest'ultimo deve necessariamente conoscere due parametri:

a) Il numero delle batterie (canali di misura) previste.

b) La soglia di allarme sulla misura effettuata (valida per tutti i canali).

Il punto a) è evidente: la scheda SLAVE deve "sapere" quanti ingressi sono previsti nel suo hardware. Fornendo un numero di ingressi superiore a quello reale, il sistema interrogherà canali inesistenti è andrà in allarme per superamento della soglia non appena il ciclo delle misure interesserà uno di questi. Invece immettendo un numero di ingressi inferiore, il sistema non darà alcun avvertimento immediato ma, in compenso, non fornirà le misure delle batterie che abbiamo "dimenticato". E' quindi importante una corretta impostazione di questo dato che, nel nostro progetto, rappresenta il punto d'incontro più importante del binomio hardware e firmware. Il punto b) è relativo alle soglie di allarme sulle tensioni misurate che, ovviamente, sono superate solo se scendono sotto un certo valore di tensione. E' proprio quest'ultimo parametro che l'utente dovrà decidere, esso sarà scelto fra una gamma di 16 valori distanziati di 0,2 V. La tabella 4 dovrebbe

aiutarci, con qualche considerazione, a trovare un compromesso per la nostra scelta.

Valori della tensione di soglia allarme con descrizione dell'applicabilità	
Valore in Volt	**Condizioni teoriche di applicabilità**
9,6	Soglie molto permissive che provocheranno un allarme solo con batteria in grado di fornire un residuo energetico molto basso. Da usarsi in casi molto particolari (attesa manutenzione, batteria difettosa, ecc.).
9,8	
10,0	
10,2	Soglie moderatamente permissive da usarsi con cautela se l'impianto prevede richieste di energia molto protratte nel tempo.
10,4	
10,6	Soglie da scegliere per un impianto moderatamente efficiente e con richieste d'energia regolari.
10,8	
11,0	
11,2	Soglie normali nel caso si prevedano sporadiche cadute della rete luce. E' questa la scelta consigliata per la maggior parte degli impianti privati.
11,4	
11,6	Soglie moderatamente severe ma necessarie in caso l'energia fosse richiesta per applicazioni professionali ed ospedaliere.
11,8	
12,0	Soglie molto severe che prevedono sicuramente un allarme nel caso di cadute di rete luce appena prolungate.
12,2	
12,4	Soglie da usarsi solo con batterie nuove e con efficienza al 100% o durante verifiche tecniche dell'impianto.
12,6	

Tabella 4

Consiglio vivamente di non scegliere soglie troppo alte perché, se la rete luce mancherà per qualche ora, con batterie mediamente efficienti l'allarme è sicuro. Una volta immessi, i dati saranno memorizzati nella EEPROM dell'MCU che, come ben sappiamo, è una memoria "non volatile". Ne consegue che, anche togliendo l'alimentazione al sistema, quest'ultimo sarà in grado poi di "ricordare", una volta riavviato, i parametri forniti dall'utente. Al momento della nuova accensione, i dati del sistema memorizzati saranno presentati all'utente che dovrà decidere se confermarli o aggiornarli. In altri termini, il programma esegue un flusso come quello mostrato, opportunamente semplificato, in figura 10.

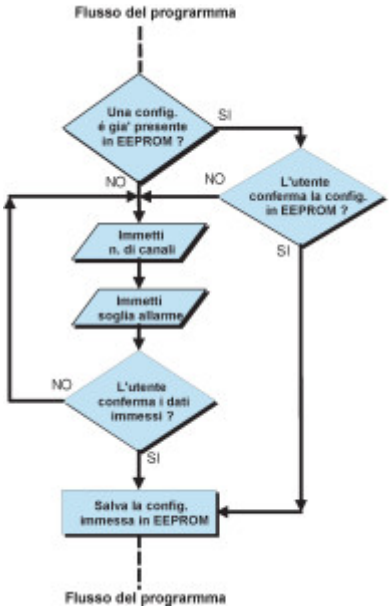


Figura 10: Diagramma di flusso dell'immissione dei dati

Durante le operazioni di immissione dati, la retro-illuminazione rimarrà sempre attiva per poi spegnersi soltanto dopo la ricezione della prima misura. Comunque, alla prima accensione assoluta, dopo la consueta visualizzazione delle versioni firmware MASTER e SLAVE, apparirà la scritta mostrata in figura 11 A.

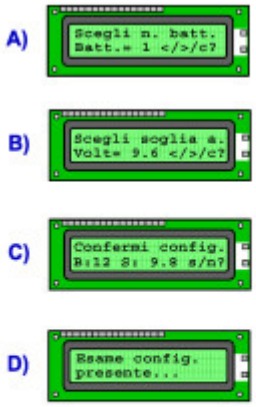


Figura 11 A-B-C-D: Visualizzazione di suggerimenti per l'immissione dei dati

L'utente, prima di avviare il sistema di misura per la prima volta, dovrà immettere sia il numero di batterie che la soglia d'allarme scelta. Questa operazione è necessaria poiché non esistono dati precedentemente memorizzati nella EEPROM. L'immissione sarà fatta utilizzando i 3 tasti e seguendo i suggerimenti del display evidenziati, per semplicità, da un '?' finale. Osservando con attenzione la figura 11 A, noteremo come la riga superiore del display ci suggerisce l'argomento della nostra scelta: 'Scegli n. batt.' (abbreviazione della frase: 'Scegli numero batterie'). In questa fase, l'utente deve immettere un numero da 1 a 16 che "dirà" al firmware quanti sono i canali di misura previsti nell'hardware. La prima scelta visualizzata sarà ovviamente rappresentata dal

numero più basso di canali: ‘1’, un’opzione un po’ pleonastica visto che impianti composti da una sola batteria sono molto rari. L’utente sceglierà il numero di canali desiderato facendo scorrere i valori presentati con i tasti freccia destra o freccia sinistra e poi confermando con il tasto <RET>. Nel programma questi tasti sono definiti rispettivamente TASTO_SI, TASTO_NO e TASTO_CONF. Questa denominazione ha la sua logica in quanto i simboli grafici non sono riconosciuti dal compilatore. Per inciso, il tasto freccia destra è usato talvolta nel programma come ultima conferma di sicurezza dell’utente. Durante la visualizzazione delle opzioni presentate, la pressione del tasto freccia destra incrementerà i valori di scelta, viceversa avverrà invece per il tasto freccia sinistra. Essendo la presentazione dei dati fornita da una matrice “circolare”, incrementando oltre il massimo numero di canali consentito (‘16’) ci troveremo di nuovo alla scelta relativa ad un solo canale. E naturalmente ciò accadrà anche decrementando il numero di batterie (pressione del tasto freccia sinistra) rispetto alla scelta visualizzata. In altri termini, la pressione ripetuta di un tasto qualsiasi presenterà comunque tutti e 16 i valori, partendo dal basso o dall’alto. Una volta immesso il numero di canali previsto, ovvero il numero di batterie dell’impianto, e data la conferma con il tasto <RET>, il display presenterà un testo simile a quello raffigurato in figura 11 B (vedi). E’ questo il momento di impostare il valore della soglia di allarme, essa sarà scelta fra le 16 possibili. Anche in questo caso, l’inizio dei valori presentati parte dalla soglia più bassa: ‘9,6’ Volt (vedi tabella 4 per maggiori dettagli). Noi effettueremo la scelta con un iter identico a quello già visto per l’impostazione del numero di canali, non mi dilungherò quindi in ulteriori spiegazioni. La conferma della scelta del valore di soglia ci porterà alla schermata di figura 11 C. In essa, saranno riepilogate le scelte effettuate e richiesta un’ulteriore conferma da parte dell’utente. Quest’ultima dovrà essere fatta, come accennato, con il tasto freccia destra (SI). Il riepilogo delle scelte appare nel rigo inferiore del display con le abbreviazione (ahimé necessarie…) di ‘B’ per numero batterie e ‘S’ per il valore di soglia scelto, quest’ultimo espresso in Volt. Il rifiuto dei dati proposti sarà fatto con il tasto freccia sinistra (NO), premendolo il programma si riporterà all’inizio dell’intera procedura di scelta. Se i parametri sono stati accettati dall’utente (vedi anche figura 10), il firmware li salverà nella EEPROM. Essendo questa una memoria permanente, alle successive accensioni del sistema i dati saranno nuovamente presentati. L’utente dovrà quindi scegliere se mantenere la vecchia configurazione, ed in questo caso salterà la procedura di scelta, o immettere dei nuovi dati aggiornati. L’intera opzione sarà presentata inizialmente come mostrato nella figura 11 D per poi passare, dopo circa 2 secondi, alla scritta mostrata in figura 11 C. Anche in questo caso la conferma dei dati presenti in EEPROM sarà fatta ricorrendo al tasto freccia destra (SI). Il rifiuto degli stessi, comporterà una nuova procedura d’immissione nelle modalità già esaminate. E’ interessante notare che eventuali cambiamenti della configurazione impostata possono avvenire solo riavviando il sistema, ovvero con un RESET generale. L’impostazione dei dati dell’impianto, tra l’altro, è una condizione assolutamente necessaria per procedere con le letture dei canali. In altre parole, non esistono valori di default che il firmware potrà considerare evitando la procedura di configurazione fatta dall’utente. Vediamo ora come è possibile interpretare i dati di misura visualizzati sul display. Cominciamo con il dire che, per conoscere tutte le misure delle batterie componenti la serie, l’utente dovrà osservare un ciclo completo. Infatti, ogni misura rimane sul display pochi secondi per poi essere rimpiazzata da una nuova effettuata sul prossimo canale. Questa limitazione, com’è evidente, è imposta dalle poche righe del nostro display (solo due!) che non permettono la visualizzazione contemporanea di più misure di canale. La tensione totale della serie di batterie, ottenuta calcolando il totale di tutte le singole misure, è invece sempre visualizzata sul display ma sarà aggiornata soltanto alla fine di un ciclo completo. Se quest’ultimo ha subito interruzioni e/o errori nel flusso dati, la visualizzazione mostrerà un ‘?’ ed il totale serie non sarà mostrato perché incalcolabile. Nella figura 12 A-B-C troviamo tre esempi di visualizzazione di misura che chiariranno quanto esposto.

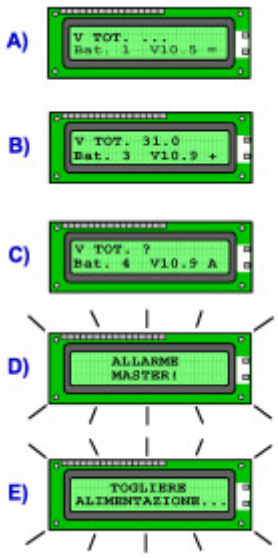


Figura 12 A-B-C-D-E: Esempi di visualizzazione LCD

Nella figura 12 A abbiamo la situazione che presenterà il display al primo ciclo di misure, quando, non essendo terminato quest'ultimo, la misura della tensione totale di serie non è ancora disponibile (nella fattispecie è sostituita dall'indicazione: '...'). Ovviamente la misura del canale corrente è visualizzata (nel nostro esempio il canale di misura '1' che ha una tensione pari a '10,5' Volt) ma, mancando termini di confronto con la misura precedente, il simbolo indicante la tendenza è di default un '='. Ricordiamo che questi può essere anche un '+' o un '-' se la lettura attuale è, rispettivamente, maggiore o minore della precedente. Nella figura 12 B ci troviamo invece all'interno di un ciclo di misure con il sistema già ampiamente avviato. Notiamo che appare sia valore della tensione totale serie, sia l'attributo di misura indicante che la misura attuale è cambiata apprezzabilmente da quella precedente (nel nostro caso in più). Un punto interrogativo al posto della misura della tensione totale, così come appare nella figura 12 C, ci indica invece che è avvenuto un errore di trasmissione, ovvero la misura di uno o più canali non è stata ricevuta correttamente. Nella videata è comunque presente il valore della tensione rilevata all'ingresso '4' con un attributo 'A'. Quest'ultimo ci avverte che la misura attuale è inferiore alla tensione di soglia impostata ed il relè d'allarme è attivo. Niente paura: la condizione è reversibile! Essa sparirà, ed il relè di allarme tornerà a riposo, non appena un intero nuovo ciclo di letture sarà compiuto senza che alcun canale riporterà misure sotto la tensione di soglia. La presenza del carattere '?' al posto della tensione totale serie, come accennato, indica che il ciclo presenta anche soltanto un'anomalia su un solo canale di misura (ad esempio: una misura non pervenuta). Infatti, in questo caso, il firmware non può calcolare una tensione totale corretta ed aggiornata poiché non conosce le tensioni di ogni singola batteria. Per avviare la retro-illuminazione del display, ed ottenere in tal modo una visione accettabile anche in condizioni di luce ambiente carente, l'utente deve solo premere il tasto <RET>. Questo tasto (chiamato nei commenti del programma anche tasto 'CONFERMA' e nelle definizioni TASTO_CONF) dovrà essere mantenuto premuto per qualche secondo, fino all'accensione del display. La retro-illuminazione si spegnerà comunque automaticamente, questo accadrà alla fine del ciclo di misure seguente a quello in corso. Esaminiamo ora gli eventuali messaggi di allarme che potremmo leggere sul display dell'unità SLAVE. Prima, però, consentitemi di fare una precisazione. Molto spesso, specie nel settore hobbistico, si vedono listati di programmi ed applicazioni che non prevedono circostanze di malfunzionamento e/o condizioni di errore indotte dall'esterno. Tutto funziona se va tutto bene ma in caso d'errore, sia esso umano o tecnologico, non esistono procedure o protezioni che possono garantire il ripristino delle funzionalità normali o, perlomeno, limitare i

danni. Questo aspetto, a mio parere, spesso costituisce una netta linea di demarcazione fra applicazioni hobbistiche e professionali. Nel nostro progetto abbiamo cercato, nei limiti del possibile, di prevedere tutte le condizioni che possono portare ad errori di lettura e guasti da circostanze esterne (scariche elettriche, errori di cablaggio, ecc.). Tra l'altro, come accennato, il sistema è basato su una comunicazione asincrona seriale e molte risorse del firmware sono dedicate all'analisi dell'esattezza dei dati scambiati. Con questo non voglio dire che eventuali miglioramenti non siano possibili, tutt'altro! Ho speso queste parole solo per mostrare che, essendo il nostro sistema uno strumento di misura, la sua affidabilità deve essere sempre massima. Il firmware divide le anomalie riscontrate in quattro categorie:

a) Batterie impianto sotto la soglia di allarme

b) Errori di trasmissione/ricezione dati

c) Avviso di batteria di back-up in scarica

d) Guasti hardware ed extra-tensioni sugli ingressi di misura

Mentre per le categorie a), b) e c) il sistema può, se le condizioni mutano, riprendere la normale funzionalità, questo non avviene per la condizione d). Infatti, determinate situazioni riscontrate sull'unità MASTER possono portare al blocco totale e continuato del sistema di misura. Nel corso della terza puntata abbiamo accennato agli eventi che possono provocare un cosiddetto allarme "catastrofico" (caso d, visto sopra). Essi sono relativi a guasti hardware dell'unità MASTER o situazioni potenzialmente pericolose agli ingressi di misura (inversioni di polarità e sovratensioni). Da un allarme catastrofico il sistema non può "riprendersi" con mezzi propri e sarà necessario un intervento umano di ripristino. In particolare, dovremmo (in ordine d'esecuzione):

1. Togliere l'alimentazione generale e l'alimentazione di back-up (batteria).
2. Isolare tutti gli ingressi di misura della scheda MASTER attestati sul connettore J2.
3. Verificare se la scheda MASTER riporta danni evidenti, se non è possibile fare una diagnosi avviare un test ed identificare il guasto (se presente).
4. Se l'hardware non mostra guasti, riavviare il sistema. E' possibile che sia stata un'occasionale sovratensione a provocare un allarme.
5. Se l'allarme accade di nuovo, isolare uno alla volta gli ingressi di misura e riavviare il sistema fino a scovare quale batteria (o collegamento) provoca il problema.

Un allarme irreversibile è comunque visualizzato dal display con il lampeggiamento della retroilluminazione ed due scritte d'avvertimento che appariranno in alternanza (vedi figura 12D e 12E). Contemporaneamente sarà attivato il relè di allarme in modalità intermittente. Le varie condizioni di allarme, siano esse "superabili" oppure irreversibili, sono comunque riassunte in dettaglio nella tabella 5. In quest'ultima sono anche descritte le varie modalità di avvertimento che il sistema userà per informare l'utente.

Modalità di visualizzazione e azionamenti d'allarme in condizioni di anomalia				
Condizione di anomalia riscontrata dal sistema	Display	Relè di allarme	LED batteria	Note

Time-out sulla ricezione dati dall'unità MASTER	Lampeggia con scritta di avvertimento	Non è attivato (rimane nello stato precedente)	Indifferente	La condizione persiste a tempo indeterminato o finché non è ricevuta una stringa valida dall'unità SLAVE
Errore di trasmissione/ricezione	Visualizza un '?' nel campo dedicato alla misura della tensione totale	Non è attivato (rimane nello stato precedente oppure segue l'esito delle misure)	Indifferente	La condizione persiste a tempo indeterminato o finché non è ricevuta una stringa priva di errori dall'unità SLAVE, le misure dei vari canali sono comunque visualizzate
Una o più batterie sotto la soglia minima di tensione impostata dall'utente	Si accende e visualizza una 'A' nel campo attributi del canale ove la misura è sotto la soglia	Attivato in modo continuo	Indifferente	La condizione persiste a tempo indeterminato o finché tutte le misure ritornano sopra la soglia impostata
Allarme per bassa tensione su batteria di back-up presente nell'unità SLAVE	Nessuna indicazione	Non è attivato (rimane nello stato precedente oppure segue l'esito delle misure)	Acceso con cicli ON-OFF di circa 1 secondo	La condizione persiste a tempo indeterminato o finché la tensione della batteria non ritorna a valori normali
Allarme per guasto catastrofico sull'unità MASTER	Lampeggia con scritte di avvertimento che si avvicendano	Attivato con cicli ON-OFF di circa 1 secondo	Indifferente	Il sistema è fermo e l'utente deve riavviare manualmente dopo la rimozione del problema

Tabella 5

Faccio notare che ulteriori precisazioni su tutti gli argomenti riguardanti il firmware sono presenti negli abbondanti commenti sul programma sorgente Slave.c. e nel suo header slave_standard.h. Questo naturalmente vale anche per il firmware dell'unità MASTER, in questo caso i files saranno rispettivamente Master.c e master_standard.h. La filosofia con cui è stato realizzato il sorgente del programma è molto semplice: facilitare al massimo coloro che vorranno modificarlo e rendere veramente fruibile la parte didattica dello stesso.

Taratura e test dell'unità SLAVE

Anche il firmware dell'unità SLAVE, così come accennammo nel caso del MASTER, consente un test guidato dell'hardware. Per accedere a questa modalità, molto comoda in quanto tutte le "periferiche" (mi si passi il termine…) vengono provate, è sufficiente collegare l'unità ad un terminale (o PC in emulazione). Naturalmente useremo la stessa sorgente di alimentazione destinata al sistema di misura completo. Anche la batteria di back-up dovrà essere collegata ai morsetti dedicati, essa dovrà avere una carica non inferiore al 75% circa in quanto il test ne prevede la misura. E' consigliabile fare attenzione alla tensione fornita da eventuali alimentatori non stabilizzati tipo plug-in poiché, molto spesso, i valori di targa non coincidono con la tensione effettiva in uscita. Questa potrebbe dipendere eccessivamente dalle condizioni di carico che, nel nostro caso, possono avere variazioni anche del 60% durante il funzionamento normale. L'ideale sarebbe usare un alimentatore tipo switching, veda comunque il lettore di fare un buon compromesso fra economia ed affidabilità (privilegiando quest'ultima, se possibile…). In caso di dubbi, verifichiamo la tensione d'ingresso alla scheda in ogni condizione di funzionamento. Come accennato, essa non deve mai superare il massimo di 12,5 Volt o scendere sotto 10,5 Volt. La figura 4 chiarirà ogni dubbio circa i collegamenti ed i componenti necessari per il test della scheda MASTER. Si tenga comunque presente che occorrerà costruire un piccolo circuito d'interfaccia (in figura 4 è evidenziato da un

tratteggio rosso). L'unità entrerà in fase di test se, al momento di fornire alimentazione, il ponticello B1 (vedi schema elettrico in figura 3) è chiuso. Portare invece a massa il ponticello B1 durante il funzionamento normale, ovvero a fase di RESET del microcontrollore superata, non causerà alcunché. Durante il test, il terminale guiderà le operazioni dell'utente con scritte chiare ed esaustive che appariranno di volta in volta. E' anche prevista la regolazione del contrasto del display, essa sarà eseguita ruotando il trimmer RV1 ed osservando un'apposita maschera visualizzata all'uopo. Una volta che entrambe le unità MASTER e SLAVE sono stata collaudate a dovere è possibile fare una prova preliminare del sistema collegando le citate unità. Il link potrà essere fatto con lo stesso cavo seriale che abbiamo usato per il test con il terminale (cavo diretto terminato con maschio a 9 poli sul lato SLAVE e femmina a 9 poli sul lato MASTER). Se anche questa prova è superata, possiamo installare il tutto nell'ambiente operativo.

Eventuali migliorie e adattamenti

Com'è mia abitudine, al termine dell'esposizione del progetto, spendo qualche parola per coloro che vorranno ampliare e migliorare le caratteristiche di quanto presentato. Del protocollo di trasmissione abbiamo già ampiamente detto nel corso della terza puntata, rimando quindi i lettori a quelle pagine. Vista la scomodità di cablare cavi, potrebbe essere interessante rendere l'applicazione wireless. In tal caso, oltre ad un'appropriata interfaccia RF, occorrerà prevedere un'alimentazione di back-up anche sul MASTER. Un adattamento sostanziale potrebbe richiedersi nel caso fossero usate, nell'impianto sotto controllo, batterie con tensione nominale diversa da 12 V. E' infatti possibile trovare impianti con batterie da 24 Volt, 6 Volt ed anche a 2,4 Volt. In tutti questi casi, le modifiche dovranno interessare sia l'hardware che il firmware. Nei primi 2 casi (tensioni d'ingresso pari a 24 e 6 Volt) la filosofia di progetto del front-end non cambierà molto: esso sarà sempre un attenuatore seguito da un buffer (vedi figura 1 della terza puntata). Quello che dovrà cambiare necessariamente è il rapporto di attenuazione introdotto dal partitore d'ingresso. Ricordiamoci che la funzione del blocco di condizionamento del segnale è quello di adattare le escursioni della tensione d'ingresso a quelle accettabili dal convertitore A/D e di fornire, all'ingresso di quest'ultimo, una tensione prodotta da un generatore avente impedenza non superiore a circa 1,2 Kilohm. La prima affermazione ci impone di non superare la tensione di riferimento scelta, ovvero 3,4 Volt e non scendere sotto lo 0 Volt (riferimento negativo del convertitore A/D). Il valore di tensione di 3,4 Volt sarà il nostro fondo scala di misura, mentre ampiezze superiori devono essere viste come fenomeni di sovratensione. Riassumo quindi le modifiche da effettuare all'hardware dell'unità MASTER in caso fosse necessario analizzare impianti con batterie aventi le tensioni sopra accennate:

a) Adattamento del rapporto di attenuazione del front-end (ovvero ritocco del valore dei resistori R20, R21 ed R22 nello schema elettrico in figura 5 della terza puntata).

b) Eventuale ritocco dei valori del partitore posto all'ingresso del rilevatore di sovratensione (R18 ed R19 nello schema elettrico in figura 5 della terza puntata) e del resistore di limitazione corrente posto sul foto-accoppiatore (R14 nel medesimo schema).

c) Adattamento del valore dei diodi ZENER di protezione posti in parallelo al bus di misura (B8, D9 e D10 nello schema elettrico in figura 6 della terza puntata).

d) Per misure su batterie a 6 Volt i valori dei resistori da R23 a R55 (vedi schema elettrico in figura 6 della terza puntata) dovranno dimezzare il loro valore. Nel caso la tensione di targa delle batterie fosse di 2,4 Volt, tale valore deve ridursi ad 1/4. Questo per consentire una sufficiente corrente di auto pulizia dei contatti dei relè.

Nel caso le batterie avessero tensione di targa pari a 2,4 Volt, il blocco di condizionamento della

tensione in ingresso dovrà avere un rapporto di attenuazione molto ridotto. Infatti, l'escursione del segnale di misura è quasi perfettamente "sovrapponibile" a quello permesso dal nostro convertitore A/D (da 0 a 3,6 Volt l'ingresso e da 0 a 3,4 Volt il convertitore). Comunque, non essendo possibile in questa sede affrontare esaustivamente il discorso della conversione A/D, rimando i lettori ad altre pagine di FE ove l'argomento è trattato nella sua giusta considerazione Per avere un'idea più precisa dei parametri di tensione da osservare nelle varie eventualità, possiamo consultare la tabella 6.

Soglie di tensione previste per batterie aventi tensione di targa diversa da 12 V				
Tensione batteria	Range normale di misura	Misura fuori scala (senza alcun allarme)	Allarme sovratensione	Clamp dei diodi di protezione posti sul bus di misura
24 Volt	Da 0 a 36 Volt	Da 36 a 46 Volt	Al di sopra di 46 Volt	Al di sopra di 47,5 Volt
6 Volt	Da 0 a 9 Volt	Da 9 a 13,5 Volt	Al di sopra di 13,5 Volt	Al disopra di 14,5 Volt
2,4 Volt	Da 0 a 3,6 Volt	Da 3,6 a 4,5 Volt	Al di sopra di 4,5 Volt	Al di sopra di 5,5 Volt

Tabella 6

Per quanto riguarda il firmware, le modifiche potranno interessare sia il MASTER che lo SLAVE (il secondo in misura maggiore). In prima approssimazione, direi che sono migliorabili alcuni aspetti relativi alle opzioni a scelta dell'utente e forse qualche informazione in più sulla tensione totale serie. La struttura del programma, in entrambe le schede, potrà comunque rimanere invariata. Sono come sempre a disposizione di coloro che vorranno intraprendere questa "avventura", consigliata solo ai lettori con una minima esperienza di programmazione in linguaggio C. Termino il discorso delle modifiche accennando ad una "ghiotta" possibilità che offre questo sistema di misura. Eccola: risparmiando la costruzione dell'unità SLAVE, la scheda MASTER può essere interrogata da un PC su cui è implementato un apposito programma. In questo caso la visualizzazione delle misure può essere decisamente migliorata (un canale può rimanere visualizzato anche per la durata di un intero ciclo) ed è possibile implementare funzionalità grafiche e statistiche. A buon intenditor, come sempre, poche parole…

Conclusioni

Terminiamo qui la trattazione di un progetto che, pur sembrando a prima vista abbastanza complesso, è in realtà alla portata del costruttore medio. Consiglio il lettore che vorrà cimentarsi di leggere bene i commenti del listato sorgente. Essi, anche senza mettere in conto modifiche al firmware, hanno un valore didattico che permetterà di comprendere bene l'interazione profonda fra firmware e hardware. Se alcune soluzioni costruttive non sono state dettagliate a fondo, lo si è fatto per lasciare al lettore la massima adattabilità di questa realizzazione alle condizioni dell'impianto. Consiglio comunque, se possibile, il montaggio su circuito stampato e l'uso di componenti strettamente attinenti alle liste parti. Non dimentichiamo, lo ribadisco per l'ultima volta, che stiamo costruendo un sistema di misura.

Bibliografia

- C Compiler Reference Manual Giugno 2008 (CCS Incorporated)
- Analisi e programmazione strutturata (Jean Pierre Laurent – Tecniche Nuove)
- PICkit 3 Programmer/Debugger User's Guide (2010 MICROCHIP – codice documento:

DS51795B)

- Le configurazioni elettroniche dei generatori di corrente costante (M. Montanari – www.fieldsforlife.org)

- Pilotare LCD alfanumerici (quinta puntata, A. Giannico) numero 331 di FE

L'evento verticale di riferimento
Tecnologie • Soluzioni applicate • Formazione

SAVE

Mostra Convegno delle Soluzioni
e Applicazioni Verticali di Automazione,
Strumentazione, Sensori.

Fiera di Verona
28-29 ottobre 2014

Sponsored by
auma
Solutions for a world in motion

Supported by
GISI
Associazione Imprese Italiane di Strumentazione

- ✔ Sistemi di controllo (DCS, PLC, PC industriali)
- ✔ Strumentazione industriale di misura e controllo
- ✔ Interfacciamento HMI
- ✔ Sensoristica
- ✔ Quadri e regolatori
- ✔ Software per l'industria
- ✔ SCADA, telecontrollo e reti tecnologiche
- ✔ Strumentazione da laboratorio
- ✔ Valvole e attuatori
- ✔ Efficienza energetica
- ✔ Motion control
- ✔ Fieldbus e comunicazione

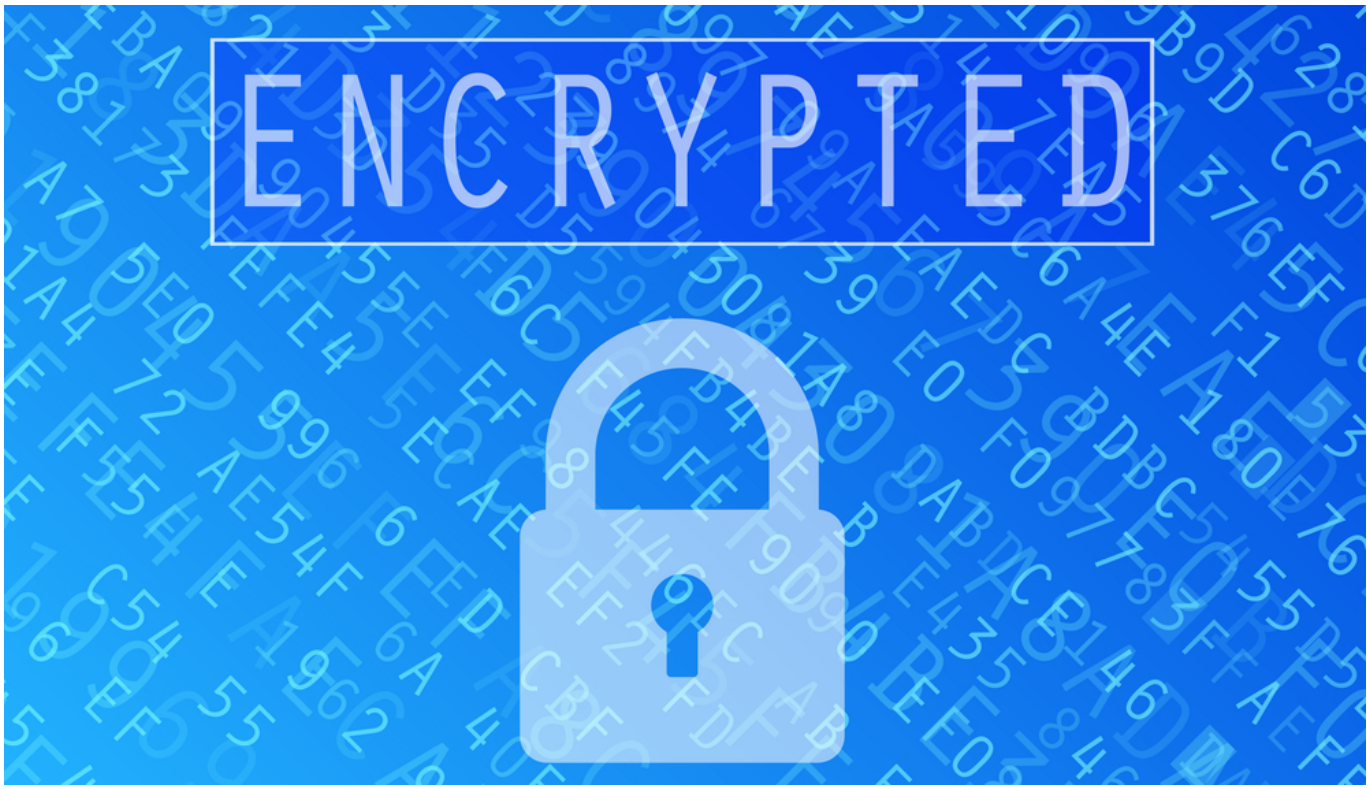
EIOM
VERONAFIERE

Il miglior rapporto qualità/prezzo
- Stand preallestiti
- Workshop tecnici
- Incontri verticali

www.exposave.com

Routine di encryption per PIC24

By Redazione | Fare Elettronica 346 – Giugno 2014



Esistono tre standards di cifratura approvati per l'uso secondo il Federal Information Processing Standards (FIPS). Nell'articolo si descrive l'implementazione di due di questi per i PIC24 e i dsPIC30/33: il Triple Data Encryption Standard (TDES) e l' Advanced Encryption Standard (AES).

TDES ENCRYPTION

Background

Lo standard originale di cifratura di dati (DES), un sistema basato su blocchi a 64-bit, è stato inventato nei primi anni '70 dalla IBM. Il DES usa una chiave di cifratura a 64-bit: 56 bits per cifrare e decifrare, i rimanenti per il controllo di parità. Fu adottato dal governo degli Stati Uniti nel 1977 come standard di cifratura di dati. A partire dalla metà degli anni '90 diverse organizzazioni pubbliche hanno dimostrato di essere in grado di craccare un codice basato su cifratura DES nell'arco di alcuni giorni Il Triple DES (TDES) è una variante del DES ed è descritto nella documentazione FIPS. Il TDES utilizza chiavi di cifratura da 56 bits a 112 o 168 bits, a seconda delle operazioni. A causa della debolezza riscontrata nell' algoritmo DES, si ritiene che la sicurezza attuale si possa basare su chiavi di 80 o 112 bits. L'uso del TDES fu suggerito dal governo americano nel 1999 in tutti gli ambiti ad eccetto di quelli legali dove solo il DES era utilizzabile. Esistono diversi modi di utilizzare il TDES; il più comune consiste nel far uso di due differenti chiavi: il dato è cifrato con una chiave e il risultato è decifrato con la seconda; si ricorre poi ancora una volta alla prima chiave. Altri modi di utilizzo fanno ricorso a tre chiavi differenti, una per ciascuna fase e alla cifratura senza avere un dato intermedio decifrato. Per la maggior parte delle nuove applicazioni, il TDES è stato sostituito con l' Advanced Encryption Standard (AES). L'AES fornisce un livello di sicurezza che è un po' più elevato rispetto al TDES ed è più veloce e pratico da implementare che il TDES. L'algoritmo originale DES è riportato in figura 1: il ciclo viene ripetuto 32 volte prima che il

testo cifrato sia valido.

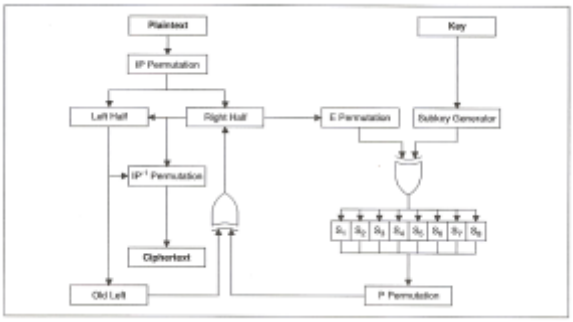


Figura 1: Algoritmo originale DES

Nell'algoritmo originale DES il testo da cifrare viene permutato da una matrice di permutazione iniziale (IP), suddiviso in una parte sinistra e in una parte destra; la parte destra viene permutata tramite una matrice indicata con E, si effettua un'operazione di XOR con una chiave opportuna sostituita tramite matrici denominate S-Box, permutate tramite matrici P; vi è un'altra operazione di XOR con la parte sinistra del dato che si giàè considerata. La parte sinistra è sostituita dalla destra precedentemente considerata e la parte destra sostuita con il nuovo valore calcolato. Il ciclo è ripeturo in modo iterativo 32 volte e il risultato è infine permutato tramite una matrice di permutazione inversa IP^{-1} per ottenere il testo cifrato finale .

Un'implementazione opzionale, il cui algoritmo è presentato in figura 2, si può considerare per ridurre il tempo di esecuzione richiesto da ogni cifratura. Dal momento che le matrici di sostituzione S-Box e quella di permutazione P coinvolgono operazioni lineari si possono combinare in una sola operazione, invece di due separate, che viene rappresentata tramite una tabella PS. Svolgere il loop DES una volta come detto elimina il dover ricorrere a variabili provvisorie e riduce il prodursi di dati in sovrappiù. Va comunque ad aumentare la dimensione del codice. Una descrizione più dettagliata delle operazioni di permutazione e sostituzione si possono trovare nella documentazione Microchip (AN583) riportata nei riferimenti.

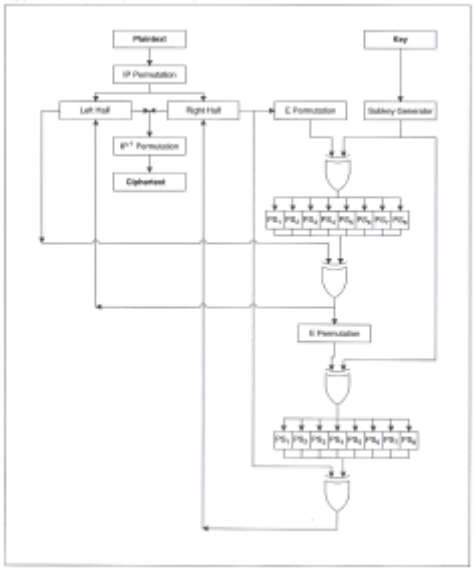


Figura 2:Algoritmo DES ottimizzato in velocità

Uso dell'algoritmo TDES

L'implementazione del TDES avviene attraverso tre funzioni chiamate: initTDES, TDES_encrypt e TDES_decrypt. Il loro utilizzo è discusso di seguito.

initTDES

Questa funzione calcola i gruppi di chiavi necessarie per il TDES; calcolando in questo modo le chiavi le routines di cifratura e decifratura vengono migliorate in termini di velocità.

Sintassi

Void initTDES (unsigned int *KeyLocation) ;

Parametri

KeyLocation: spazio degli indirizzi di memoria RAM suddiviso in word riservato alle chiavi che sono state calcolate: sono 384 byte (192 word) di area di memoria.

Valori di ritorno

Nessuno.

Dichiarazioni

KeyLocation : consiste in un' area di memoria riservata o allocata di 384 Bytes (192 words).

unsigned int Key[12] : contiene le chiavi di cifratura/decifratura, in particolare Key[0-3] è riservata alla prima chiave DES, Key[4-7] alla seconda e Key[8-11] alla terza.

Le chiavi usate per la cifratura servono per la decifratura.

Effetti collaterali

Valori ad indirizzi riservati sono cambiati.

Esempio

```
...

unsigned int  *KeyPointer;

KeyPointer = (unsigned int*)malloc(384);

if(KeyPointer  != NULL)

{

initTDES( KeyPointer);

}
```

…

TDES_encrypt

Questa funzione utilizza un insieme di chiavi generate dalla funzione initDES e cifra i dati usando l'algoritmo TDES.

Sintassi

```
Void TDES_encrypt (unsigned int *KeyLocation);
```

Parametri

KeyLocation: puntatore alla RAM dove sono memorizzate le chiavi.

Valori di ritorno

Nessuno.

Dichiarazioni

initTDES( ) viene chiamata e compare quando si calcolano le chiavi.

unsigned int M[4] viene usato con il dato che sarà cifrato.

Effetti collaterali

unsigned int M[4] viene tradotto in testo cifrato.

Esempio

…

```
TDES_encrypt (KeyPointer) ;
```

…

TDES_decrypt

Questa funzione usa un insieme di chiavi calcolate e decifra il dato usando l'algoritmo TDES.

Sintassi

```
void TDES_decrypt (unsigned iint  *KeyLocation);
```

Parametri

KeyLocation: è l'indirizzo della memoria RAM dove si trovano le chiavi Queste sono create a partire dalla stessa chiave usata a cifrare i dati (si può considerare la funzione initTDES per maggiori dettagli).

Valori di ritorno

Nessuno.

Dichiarazioni

initTDES ( ) viene chiamata e compare quando si calcolano le chiavi.

unsigned int M[4] è usato con il dato che sarà decifrato.

Effetti collaterali

unsigned int M [4] viene tradotto in testo decifrato..

Esempio

…

```
TDES_decrypt (KeyPointer);
```

…

AES ENCRYPTION

Background

Verso la fine degli anni ’90, il National Institute of Standards and Technology (NIST) iniziò a sviluppare algoritmi di cifratura che avrebbero sostituito il DES. La competizione consisteva nel testare gli algoritmi in termini di sicurezza e velocità di esecuzione per stabilire quale sarebbe stato nominato il nuovo Advanced Encryption Standard, o AES. L’algoritmo scelto , chiamato “Rijndael”, è stato adottato dal NIST nell’ottobre del 2000 e descritto nella documentazione FIPS 197. Il Rijndael/AES è un algoritmo di cifratura che fa uso di una sola chiave per cifrare i dati.L’implementazione dell’AES descritta è basata su un blocco di dati di 16 byte e una chiave di 16 byte.

Cifratura

Ci sono diverse suddivisioni di base dell’algoritmo di cifratura mostrato nella figura 3

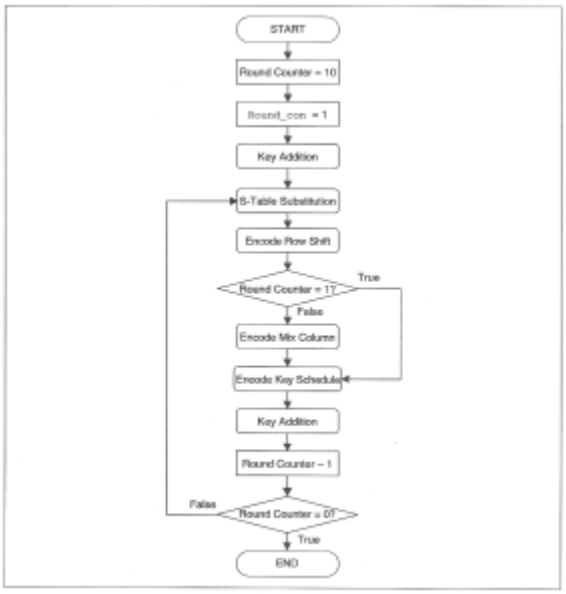


Figura 3: Diagramma di flusso dell’algoritmo di cifratura AES

Il numero di iterazioni che intervengono nell’ algoritmo si può vedere nella tabella riportata di seguito.

Dimensione della chiave	**Iterazioni necessarie in relazione alla dimensione dei blocchi**		
	16 Byte	**24 Byte**	**32 Byte**
16 byte	10	12	14
24 byte	12	12	14
32 byte	14	14	14

Tabella 1: indicazione delle iterazioni dell’algoritmo AES

Il testo da cifrare viene suddiviso in blocchi di dimensioni appropriate e inserito in un’opportuna matrice di dati; dopo aver scelto la chiave, inizia il ciclo di cifratura.

AGGIUNTA DELLA CHIAVE

Una volta che la chiave è stata scelta, si esegue un’operazione di XOR tra ciascn byte della chiave e il corrispondente byte dei dati. All’iterazione successiva la chiave generata per quelll’ iterazione e il dato subiscono un’operazione XOR a livello di byte.

SOSTITUZIONE TRAMITE TABELLA-S

A ciascuna iterazione, ciascun byte di dati è sostituito con un corrispondente byte di una tabella di sostituzione S. Una tabella simile si può vedere in figura 4.

x \ y	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
01	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
02	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
03	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
04	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
05	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
06	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
07	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
08	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
09	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
0A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
0B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
0C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
0D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
0E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
0F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figura 4: Tabella di sostituzione di cifratura S-table

SHIFT DELLA RIGA NELLA CIFRATURA

Le righe della matrice del blocco di dati subiscono una traslazione ciclica verso sinistra in modo differente come si vede in figura 5.

Before Row Shift:			
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15
After Row Shift:			
0	4	8	12
5	9	13	1
10	14	2	6
15	3	7	11

Figura 5:Tabella di shift ciclico di cifratura

TRASFORMAZIONE MIX DI CIFRATURA DELLA COLONNA

Le regole presenti nella documentazione (capitolo 2, sezione 4.2.3) dell'algoritmo AES (FIPS 197) definiscono la trasformazione chiamata mix column. In questa operazione una matrice 4×4, c(x), è moltiplicata con un vettore iniziale (a(x)) secondo le regole riportate di seguito per dare come risultato il vettore b(x) come mostrato nell'equazione di figura 6.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

FIXED MATRIX c(x)

Figura 6: Equazione 1 con matrice fissa c(x)

a•1=a

a•2=xtime(a)

a•3**=a+**xtime(a)

a•4**=**xtime(xtime(a))

a•5**=a+**xtime(xtime(a))

…

dove + è un'operazione di XOR e xtime è una procedura lineare di shift che si può descrivere in C come mostrato nell'esempio di figura 7.

```
if(a<0x80)
{
        a<<=1;
}
else
{
        a=(a<<1)^0x1b;
}
```

Figura 7: Esempio 1: Routine xtime

La prima riga della moltiplicazione risultante è mostrata nell'equazione di figura 8.

$$b[0] = xtime(a[0]) \oplus (a[1] \oplus xtime(a[1])) \oplus a[2] \oplus a[3]$$

where: "⊕" is the XOR operation

Note: The members of the multiplication are XORed together rather then added together as they would in regular matrix multiplication.

Figura 8: Equazione 2

Una dimostrazione più completa è riportata nella documentazione della Microchip riportata nei riferimenti (DS00821).

ELENCO DELLE CHIAVI DI CIFRATURA

Ciascuna iterazione dell'algoritmo AES usa una chiave di cifratura diversa basata sulla chiave precedente. L'algoritmo che le fornisce utilizza la S-table, la routine xtime e la variabile Round_con, un valore di cifratura che è inizializzato. Si considera una matrice di chiavi come quella riportato nella seguente tabella.

K0	K4	K8	K12
K1	K5	K9	K13
K2	K6	K10	K14
K3	K7	K11	K15

Tabella 2: chiave generica

Con questa matrice di chiavi si procede come segue:

1. I valori della terza colonna (da K12 fino a K15) sono usati per ottenere valori dalla tabella S.
2. La colonna 0 della matrice di chiavi (da K0 fino a K3) subisce un'operazione di XOR con i valori della colonna 3 della tabella S.
3. K0 subisce un'operazione di XOR con la variabile Round_con (il valore iniziale di Round_con è 01h).
4. Il valore di Round_con è poi aggiornato con la procedura Xtime applicata a Round_con.
5. La colonna 1 subisce un'operazione di XOR con la colonna 0.
6. La colonna 2 subisce un'operazione di XOR con la colonna 1.
7. La colonna 3 subisce un'operazione di XOR con la colonna 2.

Decifratura

Le divisioni funzionali dell'algoritmo di decifratura sono simili a quelle dell'algoritmo di cifratura essendone quasi l'operazione inversa.La differenza maggiore, comunque, è il setup che precede la decifratura. La chiave di decifratura differisce da quella di cifratura; si calcola in un certo numero di iterazioni attraverso un' apposita procedura (key schedule). Anche per l'algoritmo di decifratura si deve scegliere un valore di Roun_con, 36h, come mostrato in figura e il numero di iterazioni è 10.

AGGIUNTA DELLA CHIAVE

Come nel processo di cifratura , ciascun byte della chiave di decifratura iniziale subisce un'operazione di XOR con i bytes corrispondenti dei dati. Alla iterazione successiva la chiave generata dalla procedura di creazione subisce un'operazione di XOR a livello di byte con il dato.

OPERAZIONE MIX DI DECIFRATURA DELLA COLONNA

L'operazione inversa di mix della colonna differisce da quella di cifratura solo nella matrice di coefficienti c(x), come si vede nell'equazione 3 di figura 9.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

FIXED MATRIX c(x)

Figura 9: Equazione 3 con matrice fissa c(x)

TABELLA DI SOSTITUZIONE SI

Alle sostituzioni del processo di cifratura della tabella S seguono le operazioni che fanno uso di una tabella Si. Ad ogni iterazione ciascun byte di dati è sostituito con un byte della tabella Si, come si vede in figura 10.

		y															
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
x	00	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	10	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	20	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	30	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	40	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	50	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	60	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	70	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	80	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	90	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A0	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B0	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C0	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D0	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E0	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F0	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Figura 10: Tabella di sostituzione di decifratura Si-table

SHIFT DI RIGA DI DECIFRATURA

Come nel caso della cifratura , lo shift di riga è uno shift ciclico verso sinistra delle righe dei dati. Questa trasformazione è diversa per la cifratura e la decifratura.

ELENCO DELLE CHIAVI DI DECIFRATURA

Ciascuna iterazione dell'algoritmo di decifratura AES fa uso della stessa chiave che è stata usata per cifrare i dati. La chiave per l'iterazione successiva si può determinare dalla chiave di decifratura precedente effettuando un'operazione inversa a quella consistente nell'elenco della chiave di cifratura.

A partire dalla chiave di decifratura dell'iterazione precedente, la creazione della chiave avviene nel modo seguente:

1. Si fa un'operazione di XOR tra la colonna 3 e la colonna 2.
2. Si fa un'operazione di XOR tra la colonna 2 e la colonna 1.
3. Si fa un'operazione di XOR tra la colonna 1 e la colonna 0.
4. Si fa un'operazione di XOR tra la colonna 0 e la colonna 3 della S-table.
5. Si fa un'operazione di XOR tra K0 e Round_con.

6. Round_con è aggiornato applicando la funzione inversa xtime a Round_con: questa funzione si può definire in C come mostrato nell'esempio di figura 11.

```
if(a&0x01)
{
        a=0x80;
}
else
{
        a>>=1;
}
```

Figura 11: Esempio 2: Routine xtime inversa.

Uso dell'algoritmo AES

L'implementazione dell'algoritmo AES fa ricorso a tre chiamate a funzione: AESEncrypt, AESDecrypt e AESCalcDecKey. Il loro uso è discusso di seguito.

AESEncrypt

Questa funzione cifra un blocco di dati di 16 byte con una chiave di 128-bit facendo uso dell'algoritmo AES.

Sintassi

```
Void AESEncrypt (int  *DataBlock, const int *EncryptKey)
```

Parametri

*DataBlock: è un puntatore al blocco di 16 byte da cifrare. Il blocco di dati ha inizio ad un indirizzo di memoria pari.

*EncryptKey: è un puntatore alla chiave a 16 byte che si usa per la cifratura. La chiave si trova ad un indirizzo di memoria pari.

Valori di ritorno

DataBlock: 16 bytes di testo da cifrare all' indirizzo corrispondente al valore sono sostituiti da 16 bytes di testo cifrato.

Dichiarazioni

Nessuna.

Effetti collaterali

Nessuno.

Osservazioni

1. Il valore massimo di utilizzo della memoria stack è di 40 bytes, includendo i 4 byte dell'indirizzo di ritorno.
2. AESEncrypt richiede 2808 cicli d'istruzione (includendo CALL,RETURN e istruzione che richiedono due parametri) quando EncryptKey è memorizzato nella memoria dati. Se EncryptKey

è salvato nella memoria di programma, otto cicli d'istruzione in più sono richiesti
3. La funzione AESEncrypt è sicura da interrupt.

Esempio

```
Int block [8];

int key [8] = {  0x0100,0x0302,0x0504,0x0706,0x0908,0x0B0A,0X0D0C,0X}0F0E} ;

…

// Carica l'array block [ ] con I dati dell'applicazione

…

AESEncrypt (block, key);
```

AESDncrypt

Questa funzione decifra un blocco di 16 byte di dati facendo uso di una chiave di 128 bit tramite l'algoritmo AES.

Sintassi

```
Void AESDecrypt (int *DataBlock, const int *DecryptKey)
```

Parametri

*DataBlock: è un puntatore a 16 byte di dati da decifrare. Il blocco di dati ha inizio ad un indirizzo di memoria pari.

*DecryptKey: è un puntatore alla chiave a 16 byte da usare per la decifratura. Questa chiave non è la stessa chiave usata nella cifratura Si usa la funzione AESCalcDecKey per ricavare una chiave di decifratura da una chiave di cifratura. La chiave si trova ad un indirizzo di memoria pari.

Effetti collaterali

Nessuno.

Osservazioni

1. Il massimo utilizzo della memoria stack è 40 byte, includendo 4 byte dell'indirizzo di ritorno.
2. Se DecryptKey è salvato nella memoria dati, l'algoritmo AESDecrypt richiede 4490 cicli d'istruzione (includendo CALL, RETURN e istruzioni che richiedono due parametri). Se DecryptKey è salvato nella memoria di programma, si richiedono nove cicli d'istruzione in più.
3. La funzione AESDecrypt è sicura da interrupt.

Esempio

```
Int block[8];

int key[8] = {  0x0100,0x0302,0x0504,0x0706,0x0908,0x0B0A,0X0D0C,0X}0F0E} ;

// La chiave stabilita è caricata con la chiave di cifratura,, prima si
calcola la chiave di decifratura
```

```
AESCalcDecKey(key);
```

…

```
// Si carica block[ ] con i dati dell'applicazione
```

…

```
AESDecrypt (block, key);
```

AESCalcDecKey

Questa funzione permette di ricavare una chiave di decifratura di 128 bits da una chiave di cifratura di 128 bits.

Sintassi

```
Void AESCalcDecKey( int *Key)
```

Parametri

*Key: è un puntatore alla chiave dii cifratura a 16 byte. La chiave si trova ad un indirizzo di memoria pari.

Valori di ritorno

Key: una chiave di cifratura di16 byte è sostituita all'indirizzo indicato dal valore da una chiave di decifratura di 16 byte.

Dichiarazioni

Nessuna.

Effetti collaterali

Nessuno.

Osservazioni

1. Il valore massimo di utilizzo della memoria stack è di 6 bytes, includendo i 4 byte dell'indirizzo di ritorno.
2. AESCalcDecKey richiede 497 cicli d'istruzione (includendo CALL, RETURN e istruzioni che richiedono un parametro).
3. AESCalcDecKey è una funzione sicura da interrupt.
4. Se la funzione non è necessaria si può eliminare per salvare spazio nella memoria di programma.

Esempio

```
Int key[8] =  {  0x0100,0x0302,0x0504,0x0706,0x0908,0x0B0A,0X0D0C,0X}0F0E } ;

//  Una chiave stabilita è caricata con una chiave di cifratura, si calcola
chiave di decifratura con AESCalcDecKey(key);
```

PRESTAZIONI

Implementazioni a 16 bit degli algoritmi TDES e AES sono stati valutati su PIC24FJ128GA010, con

clock di 32 MHz (16 MIPS). I risultati sono mostrati nella tabella seguente.

Algoritmo	**Tempo di esecuzione (cicli d'istruzione)**		**Prestazioni (Kbit/s) @ 16 MIPS**	
	Cifratura	**Decifratura**	**Cifratura**	**Decifratura**
TDES(8 bytes/blocco)	6403(1)13557(2)	6403(1)13557(2)	159(1)75.5(2)	159(1)75.5(2)
AES(16 bytes/blocco)	2808	4490	729	456

Note 1: Il valore della chiave è costante per ciascun blocco e non richiede di essere calcolat nuovamente

2:Valore della chiave è calcolato nuovamente per ciascun blocco.

Tabella 3 :Tempi di esecuzione e prestazioni de idispositivi PIC24/dsPIC® per algoritmi di cifratura

UTILIZZO DI RISORSE

L'impiego di memoria degli algoritmi sono mostrati nella tabella seguente.

Algoritmo	**RAM di dati**	**Memoria di programma (Bytes)**
TDES	430(1)	7500
AES	40	3018

Nota 1: Una riduzione addizionale dell' impiego della RAM di dati si può ottenere se la chiave 1 e la chiave 3 assumono lo stesso valore. Se il codice dell'applicazione è così modificato, l'applicazione impiegherà 302 bytes.

Tabella 4: Impiego di memoria per gli algoritmi di cifratura

SINTESI

Il TDES e l'AES sono due dei soli tre algoritmi di cifratura che sono usati secondo i Federal

Information Processing Standards. Entrambi questi algoritmi si possono implementare in modo compatto ed efficiente sui PIC24 e i dsPIC30/33.

Scopo dell'articolo è introdurre il lettore agli algoritmi e al loro utilizzo pratico nelle applicazioni. Una discussione completa degli algoritmi, i loro usi e i test si possono trovare nella documentazione FIPS.

Va ricordato a proposito degli algoritmi di cifratura che nessun o di essi è sicuro; si parla di probabilità di sicurezza. Sono' anche da considerare leggi che possono alterare il codice sorgente o i prodotti che fanno uso di elementi di crittografia.

RIFERIMENTI

A. Lovich and M. Palmer, AN583, "Implementation of the Data Encryption Standard Using PIC17C42" (DS00583), Microchip Technology, Inc., 1997.

C. Gübel, AN821, "Advanced Encryption Standard Using the PIC16XXX" (DS00821), Microchip Technology, Inc., 2002.

Informazioni sui codici sorgenti riguardanti gli algoritmi AES e TDES che sono stati considerati si possono trovare sul sito www.microchip.com.

PROMOSSO DA:

assodel

Associazione Nazionale Fornitori Elettronica

18 SETTEMBRE 2014

XI EDIZIONE

BOLOGNA

POWER FORTRONIC

an assodel electronics forum

POWER FORTRONIC

L'appuntamento italiano dell'elettronica di potenza

L'elettronica di potenza è soggetta a una innovazione continua, che richiede un costante aggiornamento da parte di progettisti e operatori. Per fare il punto della situazione, l'**Electronic Community** si incontra di nuovo a Bologna, all'evento riconosciuto come il riferimento italiano per l'elettronica di potenza.

Focus su:

- Very High Power (Power Mosfet, IGBT, IGCT, IPM ecc);
- Evoluzione tecnologica (SiC, GaN ecc);
- Power Management: soluzioni analogiche e digitali;
- Motor Control;
- Passivi (Condensatori; Filtri; Magnetici; Thermal management ecc);
- Sistemi per lo storage di energia

IL POWER FORTRONIC IN NUMERI

- 10 edizioni realizzate con successo;
- Più di 60 aziende espositrici nella scorsa edizione;
- Più di 400 visitatori intervenuti durante l'edizione 2013;
- Più di 6.000 nominativi professionali in mailing;
- Una audience qualificata: il 75% dei partecipanti sono progettisti e tecnici alla ricerca di un aggiornamento sulle tecnologie per il power

IL TARGET

Tecnici, progettisti, buyer e operatori del settore

IL FORMAT

Una mostra convegno con conferenze, dibattiti e workshop pianificati da Assodel con esperti prescelti a fornire lo stato dell'arte del settore

Community - area demo per l'incontro tra domanda e offerta di fornitori e utilizzatori

Conference - un filo conduttore di interventi selezionati e dedicati alle novità di prodotto, alle applicazioni, alle case history

Educational - seminari formativi e tutorial applicativo-progettuali per tecnici

partecipazione gratuita previa registrazione

www.fortronic.it

an assodel electronics forum

FORTRONIC

IN PARTNERSHIP CON:

LUMEN CONSORZIO ELINT

Consorzio Elettrimpex
Lumen International

INFORMAZIONI / SEGRETERIA:

Tel. 02 210.111.236
marketing@fortronicforum.com

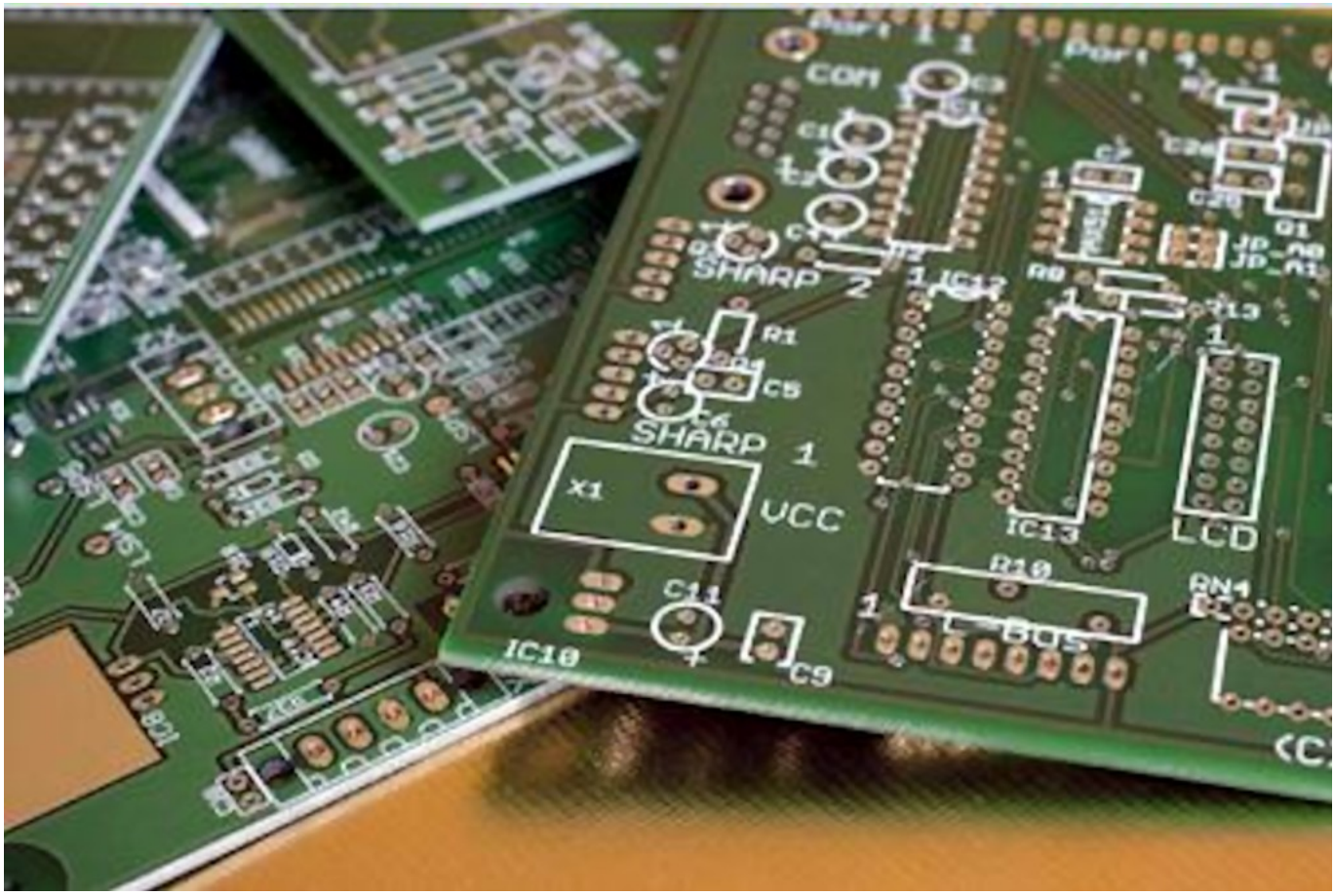
ORGANIZZAZIONE:

TECNO IMPRESE

Via Console Flaminio 19 - 20134 Milano
Tel. 02 210.111.1 - **www.tecnoimprese.it**

Fare circuiti stampati

By Redazione | Fare Elettronica 346 – Giugno 2014



Le regole progettuali da tenere in considerazione per agevolare la fase di produzione dei circuiti stampati e le possibili scelte, in termini di produttori, sul territorio nazionale e non.

Uno degli elementi di maggior peso all'interno di un progetto elettronico, hobbistico o professionale che sia, è certamente la progettazione e la realizzazione del PCB (Printed Circuit Board). Spesso si parla di progettazione, degli strumenti CAD da utilizzare, di come progettare i circuiti tenendo conto di tutte le variabili in gioco, ma quasi mai si fa riferimento al problema della realizzazione del PCB vero e proprio, ossia di quella parte di processo che, una volta terminato il progetto, porta alla realizzazione di una scheda a circuito stampato. Anche la scelta del produttore stesso del PCB può essere un interessante argomento di discussione, anche in funzione del fatto che produttori diversi si rivolgono a target diversi (prototipazione, medie forniture, produzioni in serie), ed il numero di aziende attive in questo segmento di mercato costituisce un insieme sicuramente vasto ed eterogeneo. Quest'articolo si pone l'obbiettivo di analizzare questo problema, ponendo inoltre in evidenza quali sono le regole progettuali da tenere in considerazione per agevolare il lavoro di chi poi produrrà fisicamente i nostri circuiti stampati ed analizzando quali sono alcune delle possibili scelte, in termini di produttori, sul territorio nazionale.

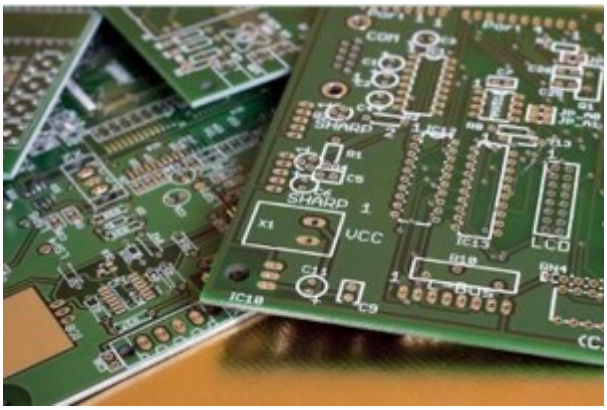


Figura 1 Esempio di circuito stampato

Figura 2 Esempio di PCB con solder mask di colore blu

Breve introduzione sui PCB

I PCB (acronimo di Printed Circuit Board, ossia “scheda a circuito stampato” o più comunemente “circuito stampato”) è il componente base di un circuito elettronico su scheda. La sua funzione è quella di fornire il supporto sul quale vengono montati i vari componenti elettronici (circuiti integrati, resistenze, condensatori, ma anche connettori, dissipatori, etc.), ed anche quella di assicurare il collegamento elettrico tra i vari componenti tramite una rete (net) di piste, solitamente di rame. In funzione della complessità del circuito e dello spazio a disposizione i PCB possono avere una o più “facce”. Ogni faccia contiene una serie di piste, ed il collegamento tra due facce è effettuato tramite i cosiddetti “vias”, ossia fori metallizzati che passano da uno strato all’altro, assicurando la continuità del collegamento elettrico. Esistono poi vari altri strati, per indicare i quali si usa generalmente il termine inglese “layer”, che compongono il PCB. Prendiamo, a titolo di esempio, un semplice PCB a due facce, che costituisce sicuramente il punto di riferimento per le realizzazioni hobbistiche. Oltre alle due facce “rame” generalmente dette lato component (destinato ad ospitare piste e componenti) e lato copper (destinato ad ospitare piste, eventuali componenti che non trovano spazio sul lato component ed anche un piano di massa) esistono infatti anche altri layer. Molta importanza rivestono il lato solder mask superiore ed inferiore.

La maschera non è altro che uno strato protettivo (storicamente di colore verde, ma si vanno sempre più diffondendo anche altre colorazioni) che serve a proteggere piste e vias da corto-circuiti accidentali o da abrasioni. Non dimentichiamo inoltre i piani serigrafie, che permettono di inserire informazioni testuali sulle schede, che possono sia servire all’identificazione di componenti durante la fase di saldatura a mano oppure contenere logi, indicazioni o qualsiasi altro tipo di informazione ritenuta utile. Esistono poi anche altri strati che assumono maggiore importanza nel caso di montaggi automatici, ma non entreremo nel dettaglio in quanto, in questo caso, ci si scosta dal

settore hobbistico e si va verso realizzazioni a livello produzione.

Regole per il disegno

Analizziamo adesso quali sono le più importanti regole da seguire durante la fase di progettazione delle nostre schede, per facilitare il lavoro del produttore e non incorrere in spiacevoli errori di realizzazione.

Isolamento Minimo

Si tratta della distanza minima che può esistere tra due piste, tra due vias o tra una pista ed un vias. Questo vincolo, se violato, potrebbe portare ad un mancato isolamento tra due elementi del PCB durante il processo di metallizzazione, compromettendo la realizzazione finale. In generale i produttori dovrebbero indicare il minimo valore tollerabile.

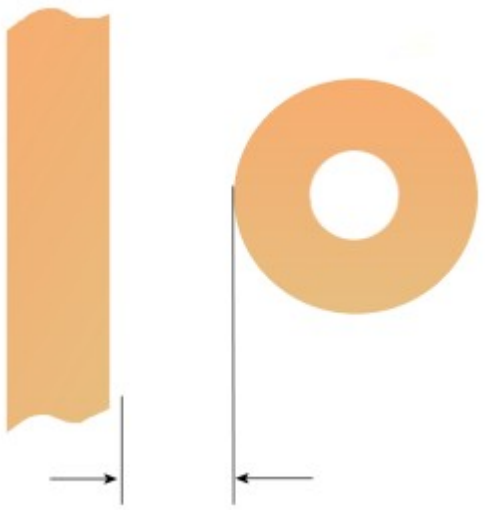


Figura 3 Isolamento Minimo

Spessore delle Piste

Ogni produttore, in dipendenza dell'attrezzatura di cui dispone, può realizzare piste fino ad un certo spessore minimo. Tale spessore minimo costituisce un vincolo di progetto che deve necessariamente essere rispettato. La violazione di tale vincolo, in genere, causa la messa in attesa dell'ordine.

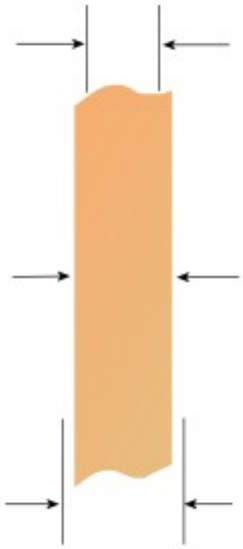


Figura 4 Spessore delle Piste

Anular ring e diametro minimo del foro

Ogni foro metallizzato (sia esso un via o una piazzola di un componente trough-hole) presenta due parametri di rilievo, che sono il diametro del foro passante e l'anular ring. L'anular ring di un foro metallizzato è lo spessore di metallo che separa il bordo esterno dal bordo interno della piazzola. Ogni produttore dovrebbe fornire come informazioni il minimo foro realizzabile dalle macchine di cui dispone e la dimensione minima dell'anular ring. La dimensione dell'anular ring (per quanto riguarda le piazzole dei componenti) incide sulla saldabilità, quindi un consiglio sicuramente valido consiste nel tenere un margine di qualche mils rispetto a quanto indicato dai produttori.

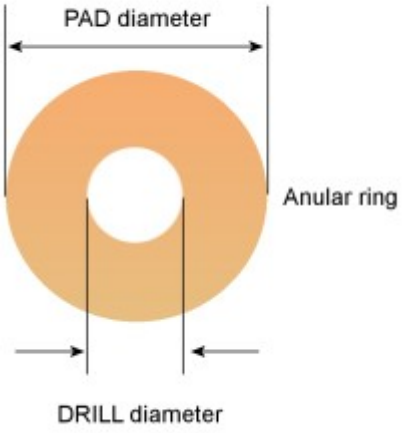


Figura 5 Anular Ring

Distanza solder mask dal rame

La maschera protettiva di solder può creare problemi nella saldatura dei componenti, nel caso invada le piazzole. Per evitare quest'inconveniente i produttori segnalano una distanza minima da

rispettare tra solder mask e rame.

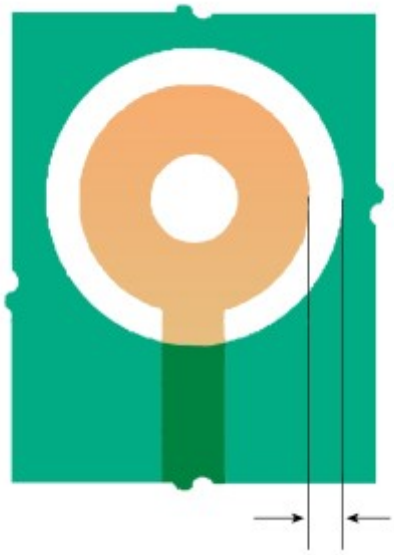


Figura 6 Distanza solder mask dal rame

Distanza serigrafie dal rame

Anche le serigrafie possono creare problemi analoghi a quelli della solder mask se invadono il rame. Vale lo stesso discorso fatto in precedenza per il solder. Ciò che abbiamo appena elencato costituisce l'insieme base di regole per il disegno, che consentono di evitare i problemi più comuni. Va da se che il disegnatore CAD acquisisce esperienza progetto dopo progetto, e che tale processo trova il suo completamento all'interno di un certo arco temporale. Le parole chiave, insomma, sono sempre esperienza e perseveranza.

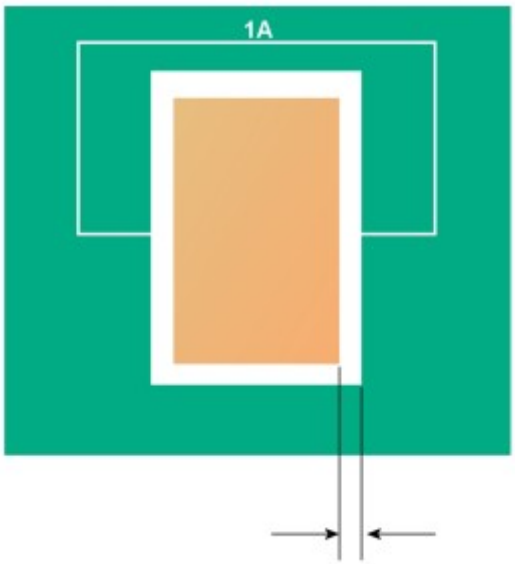


Figura 7 Distanza serigrafia dal rame

Passiamo adesso ad analizzare alcune delle aziende attive nel settore della produzione dei PCB. Naturalmente l'elenco, anche limitatamente al territorio nazionale, sarebbe molto vasto, quindi ci limiteremo a descrivere le più affermate nel settore della prototipazione e della produzione in piccole serie.

Millennium Dataware srl (MD srl)

Una delle scelte sicure. L'azienda fondata nel 1997 da Francesco Fontana opera in Italia nel settore della prototipazione e piccole serie e ha sede nella zona del parco scientifico di Rivalta Scrivia, in provincia di Alessandria. Molto indicata sia per liberi professionisti o hobbisti, che per le aziende, data la notevole perizia dei tecnici, la qualità del prodotto finale e la puntualità nelle consegne, risulta inoltre molto facile l'interfacciamento, grazie alla gestione delle commesse che avviene unicamente on-line, tramite il gestore degli ordini presente sul sito web (www.mdsrl.it). E' possibile richiedere la realizzazione di PCB singola e doppia faccia di due tipologie:

- **STANDARD**
  - Singola o doppia faccia
  - rame 35um su FR4 1,6mm
  - Finitura con Stagno Chimico Ormecon Plus
  - Fori metallizzati per i PCB doppia faccia
  - Nessuna serigrafia o solder

- **PLUS**
  - Singola o doppia faccia
  - rame 35um su FR4 1,6mm
  - Finitura con Stagno Chimico Ormecon Plus
  - Fori metallizzati per i PCB doppia faccia
  - Serigrafie e solder

Il gestore degli ordini permette di selezionare la dimensione del PCB (fino a 210×300 mm) e i tempi di consegna, che variano da 24 ore a 10 giorni per i circuiti standard e da 48 ore a 11 giorni per i circuiti della serie plus. Uno dei punti di forza dell'azienda è sicuramente l'estrema puntualità nelle consegne, che permette agli acquirenti di programmare con ottima precisione la realizzazione dei prototipi. La procedura d'ordine è estremamente semplice, dopo aver selezionato tipologia di PCB, dimensioni e tempi di consegna si carica il file contenente i gerber (vengono accettati i seguenti formati: Gerber RS274-X, Gerber RS274-D, Cadsoft Eagle .brd), compressi in formato .ZIP o .RAR e si inseriscono eventuali commenti sull'ordine. Dopodichè si sceglie il vettore per la consegna (si va dalla consegna tramite pacco postale a varie tipologie di corriere espresso). Infine si effettua il pagamento con carta di credito o bonifico bancario. Con i file già pronti e pagando con carta di credito l'intera operazione non richiede più di un paio di minuti, a testimonianza che efficienza e velocità sono entrambi paradigmi base della società. A breve sarà disponibile un novo tipo di PCB basato sulla tecnologia IMS (Insulated Metal Substrate). Tale tecnologia impiega uno strato di alluminio da 1,5mm separato dal piano rame classico (da 35 o 70um) tramite un sottile strato di dielettrico da 100um. Questo tipo di realizzazione si presta particolarmente bene all'uso con componenti che necessitano di una buona dissipazione termica ma che non possono impiegare dissipatori. L'esempio classico è costituito dai LED ad alta luminosità, che a temperature elevate subiscono un degrado di prestazioni, fornendo una minore intensità luminosa. L'alluminio di cui è composto il substrato garantisce una buona dissipazione termica, pur garantendo (grazie allo strato separatore di dielettrico) un buon isolamento.

PCB Project

Nata nel 1996, La PCB Project si afferma come produttore di circuiti stampati sia singola che doppia

faccia e si distingue, nella produzione di prototipi e piccoli lotti, per i costi competitivi ed i tempi brevi di consegna. Per offrire un sempre migliore servizio si è, pertanto, dotata, nel corso degli anni, delle attrezzature necessarie a realizzare tutte le fasi della lavorazione al suo interno in modo da adattarsi alle richieste del cliente, con flessibilità ed efficienza, sia per la produzione di serie che per il singolo prototipo. Come materiale base viene utilizzato l'FR4 ma anche CEM1, Teflon, Arlon, Roger. Gli spessori del materiale di base vanno da 0,5 mm a 2,4 mm mentre per il rame si va da uno spessore di 35µ a 105µ. Gli ordini vengono gestiti tramite il sito (www.pcbproject.it), molto ben fatto ed intuitivo. Per il calcolo del preventivo si ha a disposizione un calcolatore on-line che permette di selezionare dimensioni del PCB, numero di facce, tempi di consegna, etc.

PCB Pool

PCB Pool è un servizio di produzione PCB fornito dall'azienda Beta Layout. Beta LAYOUT è stata fondata come azienda produttrice di PCB nel 1989, e nel 1994 Arne Hofmann ebbe la brillante idea di assemblare molti ordini diversi di PCB sul pannello in modo tale che i costi di produzione di questo pannello fossero divisi fra i vari progettisti di PCB. Da quell'anno iniziò a nascere PCB Pool. I clienti di PCB Pool vanno da semplici aziende monoproprietarie e progettisti solo per hobby, a ingegneri che lavorano nei dipartimenti di sviluppo delle grandi aziende. Anche in questo caso la gestione degli ordini avviene on-line, tramite il sito (www.pcb-pool.com), molto ben realizzato e dal design accattivante. Le tipologie di prodotto sono due, prototipi e serie xxs. L'opzione prototipo è molto indicata per la realizzazione di singole schede fino a 6 layer, mentre la xxs si riferisce a piccole serie ed offre prezzi più vantaggiosi. Soldermask e serigrafie vengono trattate a parte con prezzi forfetari. Da notare che per il primo ordine sono previsti degli omaggi, una pompa dissaldante nel caso dei PCB della serie prototipi ed un multimetro palmare nel caso della serie xxs.

Alimentatori per Cellulari e dispositivi mobili

By Mario Rotigni | Fare Elettronica 346 – Giugno 2014

Tanto diffusi da sembrare invisibili, possono costituire sorgente di utili componenti e svelare problematiche insospettate

Il diffondersi di apparecchiature portatili di ogni tipo, quali cellulari, navigatori, macchine fotografiche digitali, lettori MP3, videocamere digitali, tablet, netbook, ha trascinato una tumultuosa crescita di alimentatori, caricabatterie da rete, alimentatori da auto, da prese USB e più recentemente, anche da fonti rinnovabili quali piccoli pannelli solari. A tale varietà si accompagnano le specifiche elettriche più varie e diverse prese dai molteplici schemi. Si possono quindi trovare

adattatori con meccaniche compatibili, ma con livelli di tensione assai diversi, esponendo quindi l'utente meno attento a possibili danneggiamenti degli adattatori stessi o degli apparecchi mobili collegati. Non avendo inoltre interruttore nella maggior parte dei casi, l'inserimento di questi alimentatori avviene sempre 'a caldo', con tensione applicata. Questo può comportare la generazione di transitori e sovratensioni che vanno gestite al fine di salvaguardare l'affidabilità e la sicurezza. L'Application Note indicata in rif. 1 analizza in dettaglio il comportamento al power up del sistema alimentatore da rete più dispositivo portatile.

Figura 1: alimentatori ed adattatori per apparecchiature portatili

I condensatori di uscita dell'alimentatore, le induttanze dei cavi di collegamento ed i condensatori di ingresso del carico formano un circuito oscillante RLC, in grado di produrre elevate sovratensioni nella fascia inferiore alla soglia di intervento delle protezioni tipicamente installate, ma decisamente superiori ai valori di esercizio. Il dimensionamento dei circuiti di uscita ed ingresso andrebbe quindi eseguito accuratamente, portando a soluzioni robuste e non necessariamente più costose. Come dimostrato dall'Application Note richiamata, sarebbe consigliabile collegare prima l'adattatore all'apparecchio portatile e poi inserirlo in una presa di alimentazione, riducendo il rischio di sovratensioni sull'alimentatore interno del portatile. È chiaramente impossibile garantire che questa sequenza venga rispettata sempre nella pratica quotidiana. Purtroppo non sempre il corretto dimensionamento che sarebbe necessario per far fronte a problemi viene realizzato, per varie ragioni, esponendo quindi le apparecchiature a rischi di guasto e degrado di affidabilità. Circuiti integrati dedicati a proteggere da questi rischi sono stati progettati e commercializzati, tra gli altri da Linear Technology (vedi LTC4360, LTC4361 e LTC4362). LTC4360 e LTC4361 proteggono l'elettronica a bassa tensione da sovratensioni pilotando un transistor MOSFET esterno configurato come pass transistor, sostanzialmente un veloce interruttore elettronico allo stato solido che sconnette velocemente il carico sensibile in presenza di anomalie, impedendo quindi il danneggiamento. LTC4362 ha invece il MOSFET e shunt interni, consentendo un ingombro su PCB ancora più ridotto. I datasheet di questi componenti comprendono un'ampia sezione destinata alle applicazioni ed ai benefici conseguiti, integrando di fatto un'Application Note. LTC4360 e LTC4361 sopportano fino a 85V sugli ingress e non richiedono condensatori di bypass ad alta tensione in ingresso, eliminando quindi una possibile sorgente di guasti. Il condensatore di bypass a bassa tensione richiesto in uscita coincide con il bypass del circuito di carico, non richiedendo componenti aggiuntivi.

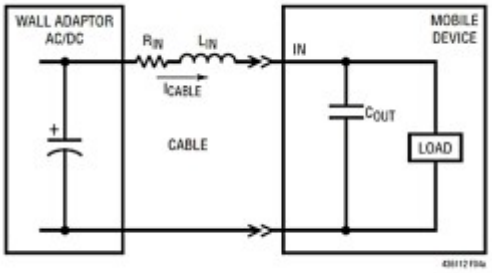


Figura 2: schema tipico di connessione tra alimentatore e carico per dispositivi portatili.

Questo condensatore lavora anche durante rapide sovratensioni, contribuendo a ridurre la variazione di tensione e dando il tempo alle protezioni integrate di spegnere il MOSFET, prima che l'overshoots raggiunga valori pericolosi. Quando viene applicata tensione, il MOSFET è inizialmente mantenuto spento da un circuito di ritardo. Il ritardo di 130msec viene triggerato nuovamente ad ogni evento di undervoltage (Vin < 2,1V) o overvoltage (Vin > 5,7V). Il MOSFET isola quindi il carico dai transitori di accensione. Allo spirare del ritardo, il MOSFET è portato in conduzione in modo controllato con una rampa lineare di 3V/msec applicata alla sua Gate e mantenuto poi in conduzione con una tensione di gate limitata per impedirne il danneggiamento. Se Vin sale oltre 5,8V il MOSFET viene spento per 130 ms. LTC4361 e LTC4362 hanno anche un circuito di protezione da sovracorrenti, leggendo la caduta di tensione dello shunt RSENSE (vedi figura 3).

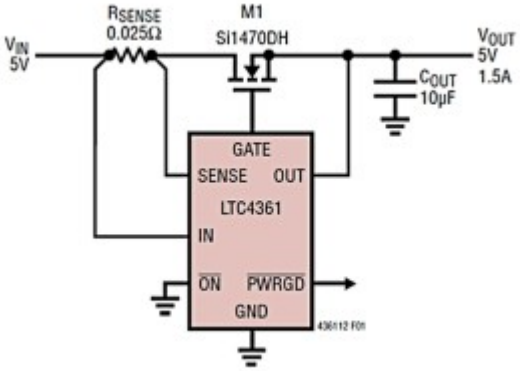


Figura 3: schema protezione sovratensione e sovracorrente per Alimentatore 5V.

Sovracorrenti possono essere generate, ad esempio, a seguito di inversioni di polarità accidentale dovuta a collegamento ad un alimentatore inappropriato. Correnti oltre 1 A causano lo spegnimento del MOSFET, anche in questo caso per un tempo di 130 ms. Un filtro anti-glitch di 10 us previene falsi interventi della protezione. Un MOSFET esterno può essere impiegato quando sia richiesto sopportare tensioni maggiori rispetto a quella garantita dal MOSFET integrato. Le figure 4 e 5 mostrano la pulizia nella tensione al carico ottenuta.

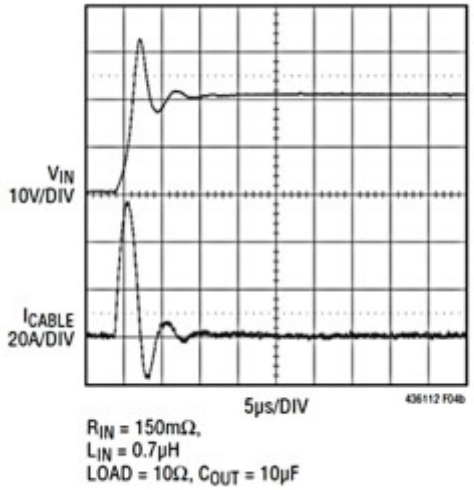


Figura 4: sovratensione in schema tradizionale.

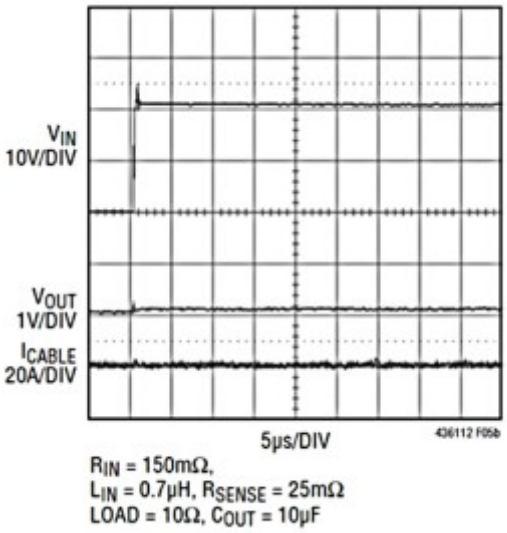


Figura 5: sovratensione in circuito con LTC4361.

Questi componenti possono essere utili anche in altre applicazioni, costituendo semplici e robuste protezioni attive da anomalie sulle alimentazioni. Particolarmente interessante è l'imposizione di una rampa di salita dell'alimentazione controllata ed abbastanza ripetibile, cosa che può creare, tuttavia, alcuni problemi inaspettati. La fase di start up dei moderni microcontrollori è certamente tra le più delicate. Il circuito destinato a fornire il clock, base di tutte le operazioni, deve innescare l'oscillazione e stabilizzarsi. Il reset interno viene rilasciato quando oscillatore e tensioni di alimentazione sono rilevate stabili ed inizia allora l'inizializzazione delle periferiche interne parte della procedura di start up del programma applicativo che prende il controllo della circuiteria esterna. Non casualmente, molta cura è stata dedicata dai produttori di microcontrollori e circuiti logici a questo argomento, vedi ad esempio rif. 2. Si capisce che un power up dell'alimentazione "pulito", relativamente ripetibile e stabile contribuisce alla robustezza dell'applicazione. In particolare è importante garantire le temporizzazioni e la monotonicità della rampa di salita, per evitare interventi intempestivi dei comparatori interni predisposti alla sorveglianza delle alimentazioni, di cui molti microcontrollori sono dotati. È importante, però, garantire che le tensioni di alimentazione siano entro specifica, prima di rilasciare il reset. Questo perché i circuiti CMOS iniziano a funzionare a tensioni molto basse, circa 2V, ben inferiori alla minima di specifica di molti

circuiti integrati (es 4.5V). Se la rampa di salita dell'alimentazione è particolarmente lenta, la circuiteria potrebbe trovarsi in una condizione intermedia, dove elettricamente è già attiva, ma l'inizializzazione non è ancora terminata ed il comportamento complessivo del microcontrollore diviene impredicibile. È importante, quindi, accertarsi che accanto a protezioni da condizioni di alimentazione anomale, i circuiti di reset siano in grado di consentire l'attivazione dei componenti solo entro valori a specifica, rispettando sia la velocità di salita minima, come intuitivo attendersi, ma non eccedendo il limite proprio del componente. Riassumendo, in presenza di alimentazioni con tempi di assestamento lunghi, è fondamentale l'uso di circuiti di gestione del reset abbastanza sofisticati da attendere che le tensioni di alimentazione siano sicuramente entro i limiti di specifica prima di consentire l'attivazione del microcontrollore.

Riferimenti

Rif.1: Linear Technology AN88 "Ceramic Input Capacitors Can Cause Overvoltage Transients".

Rif.2: Power-up Considerations Microchip AN 522, Microchip.

Scopri il nuovo ARDUINO projects

12 progetti completi

190 pagine

Solo 4.99€!!

Acquistalo ora!

PayPal MasterCard VISA

- RS232 con Arduino
- Usare Arduino con GSM/GPRS
- ArduinoBOT
- Braccio robotico
- Alla scoperta di Arduino DUE
- Scambiatore per elettropompe
- Etilometro
- Game Controller
- Lampada da tavolo intelligente
- Arduino parlante
- Controllare Arduino
- Web server

I progetti sono completi di firmware!

IE cloud

Pilotare i LED bianchi

By Redazione | Fare Elettronica 346 – Giugno 2014

In questo articolo si parlerà di LED bianchi, del perché pilotarli a corrente costante e si metteranno a confronto sei diversi LED di due marchi. L'obbiettivo e capire pregi e difetti dei circuiti di pilotaggio dei LED.

Quando si utilizzano il LED bianchi solitamente è per realizzare una retroilluminazione o una qualsiasi forma di illuminazione generica. Le ragioni per cui risulta conveniente pilotarli a corrente costante sono essenzialmente due. La prima è evitare di superare la massima corrente consentita per il LED, la seconda è ottenere un risultato di luminosità e colore prevedibile e ripetibile per ogni LED.

LED bianchi in test

Sono stati presi tre LED bianchi di un fornitore e tre LED bianchi di un altro e per tutti è stata tracciata la caratteristica corrente-tensione. Il risultato è visibile in figura 1. Pilotando i LED con una tensione di 3.4V si può vedere che la corrente varia da 10mA a 44mA a seconda del LED. Per la robustezza del sistema è importante non violare il vincolo sulla corrente massima per LED, solitamente intorno ai 30mA massimi. Tuttavia se la temperatura ambiente sale la corrente massima diminuisce come mostrato dalla caratteristica di figura 2.

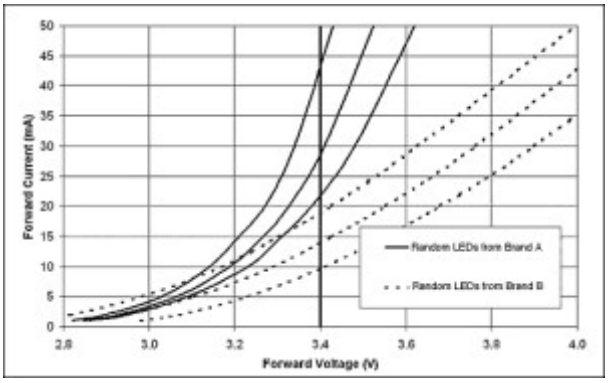


Figura 1: Corrente diretta in funzione della tensione diretta applicata al LED bianco.

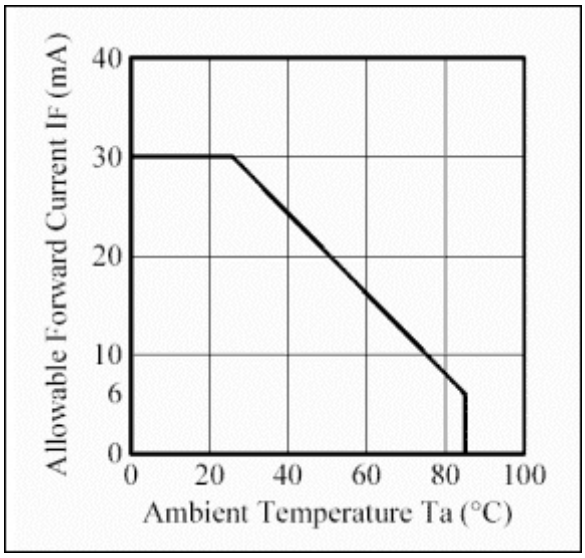


Figura 2: Corrente massima consentita per il LED in funzione della temperatura ambiente.

Se si limita la corrente massima a 20mA allora è possibile utilizzare l'applicazione fino ad una temperatura di 50°C senza rischiare che qualche LED si bruci.

È immediato desumere da quanto detto sinora che pilotare i LED a tensione costante non risulta essere una buona soluzione anche perché tipicamente il processo produttivo dei LED che prevede una verifica di luminosità e colore è realizzata con corrente costante, come si può intuire da qualsiasi datasheet.

Quatto circuiti per tre LED

In figura 3 sono presentati quattro diversi circuiti che consentono di pilotare tre LED bianchi.

Il circuito A genera una tensione costante e la regolazione di corrente è eseguita attraverso le resistenze. Il circuito B genera una corrente costante ripartita con le resistenze per ogni LED. Il circuito C è un regolatore di corrente multiplo mente il circuito D è ancora un regolatore di corrente ma con i LED in serie.

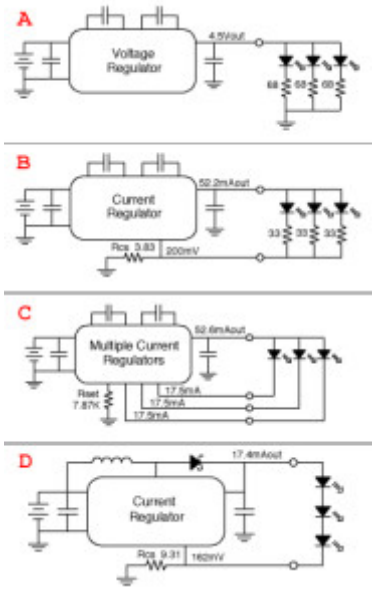


Figura 3: Quatto circuiti diversi per pilotare tre LED bianchi.

Per ognuno dei circuiti di figura 3 andiamo graficamente sulla caratteristica corrente tensione in diretta a vedere come viene eseguita la regolazione, per stabilire il punto di funzionamento.

Il circuito in figura 3-A è molto semplice ed estremamente diffuso, il suo principale pregio infatti è quello di essere realizzabile utilizzando un gran numero di regolatori disponibili in commercio. Lo svantaggio principale è legato alla sua efficienza, infatti parte della potenza viene dissipata sulle resistenze e la corrente nei LED non è controllata in maniera accurata. Dalla figura 4, relativa allo schema di figura 3A si vede che la corrente può variare da 14.2mA a 18.4mA e che i LED di marca A lavorano mediamente con 2mA di corrente in più dei LED di marca B, risultando quindi più luminosi.

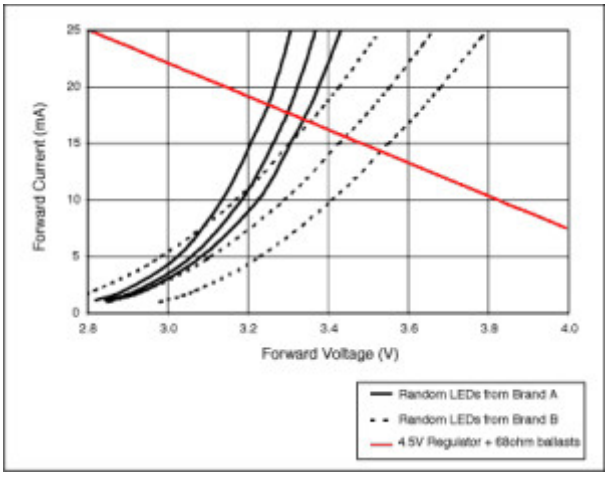


Figura 4: Punto di lavoro per il circuito di Figura 3A con i sei LED bianchi esaminati

Il circuito di figura 3B regola la corrente complessiva che circola nei 3 LED, le resistenze servono per distribuirla equamente. Come regolatore si può utilizzare per esempio il MAX1910 della Maxim Integrated Products. Il circuito proposto si basa sul principio che tra LED e LED non ci siano differenze rilevanti, almeno tra lotti ravvicinati. Le resistenze in tal caso possono essere di valore basso, riducendo così la potenza dissipata. Dal grafico di figura 5 si vede che la corrente per i sei esemplari sotto esame varia da 15.4 a 19.6mA con il marchio A che presenta comunque valori ravvicinati rispetto al marchio B. Tutti e 6 i LED sono attraversati da una corrente di 17.5mA/LED. Lo svantaggio di questo circuito è che rimane comunque una percentuale di potenza dissipata sulle

resistenze e che le correnti nei vari rami non sono mai perfettamente identiche. In ogni caso il circuito è un buon compromesso tra performance e semplicità.

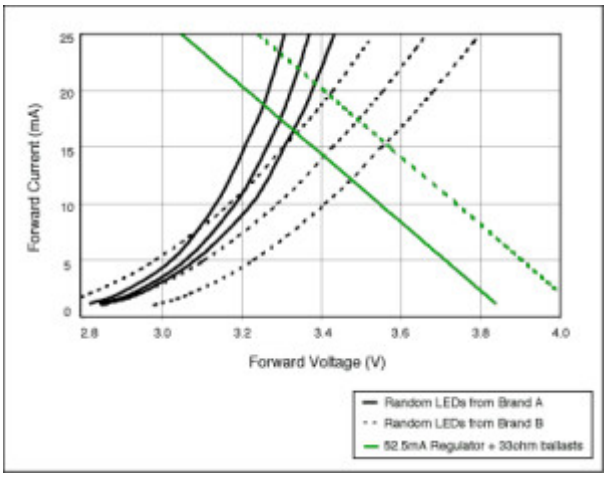


Figura 5: Punto di lavoro per il circuito di Figura 3B con i sei LED bianchi esaminati

Il circuito di figura 3C regola la corrente individualmente per ogni LED e non richiede resistenze. Il valore di corrente su ogni ramo e la ripetibilità dello stesso su tutti i rami è garantito dal regolatore, il MAX1570 sempre della Maxim Integrated Products è un tipico esempio di prodotto di questa categoria. Il MAX1570 ha una accuratezza sul valore di corrente del 2% e un accuratezza sulla ripetibilità del valore del 0.3%.

Poiché la regolazione di corrente è a low dropout, l'efficienza raggiunge valori elevati.

In figura 6 si ha il punto di lavoro per i nostri LED quando sono percorsi da una corrente costante a 17.5mA. Il circuito risulta più compatto poiché non sono necessarie le resistenze, tuttavia tra il regolatore e i LED sono necessari quatto collegamenti, con un conseguente aumento delle dimensioni del regolatore. Questo circuito garantisce alte prestazioni ad un costo però maggiore dei precedenti.

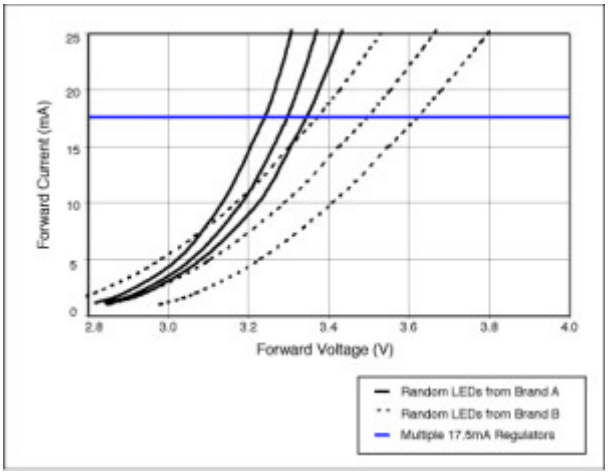


Figura 6: Punto di lavoro per il circuito di Figura 3C e 3D con i sei LED bianchi esaminati

Il circuito di figura 3D e il classico circuito boost ad alta efficienza basato su induttore, configurato per regolare la corrente. La bassa tensione di feedback minimizza la potenza dissipata nella resistenza di sense. Il punto di lavoro per il circuito di figura 3D è lo stesso di figura 6 ottenuto per il circuito di figura 3C. Poiché i LED sono in serie la corrente tra loro è esattamente identica in tutte le condizioni. La precisione della corrente è determinata dal circuito di feedback e risulta indipendente

dalle differenze sulle tensioni di forward dei LED. Il MAX1848 e il MAX1561 sono esempi di regolatori di questo tipo e garantiscono un 87% (3 LED) e un 84% (6 LED) di efficienza rispettivamente. I collegamenti tra regolatore e LED prevedono solo due connessioni e questo rappresenta un vantaggio. Lo svantaggio è rappresentato dalle dimensioni dell'induttore per il regolatore boost, dal costo e dalle EMI irradiate.

In figura 7 un circuito realizzato con il MAX1561, la corrente nei LED si può determinare dalla formula

$$I_{LED} = V_{CTRL} / (10xR_{SENSE})$$

Dove V_{CTRL} varia da 0.24V a 1.62V. Attraverso il pin di controllo del MAX1561 è possibile con una forma d'onda PWM regolare la luminosità dei LED, una variazione di duty cycle determina una variazione dell'intensità.

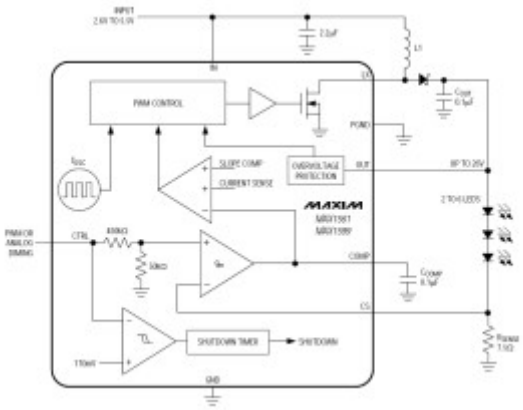


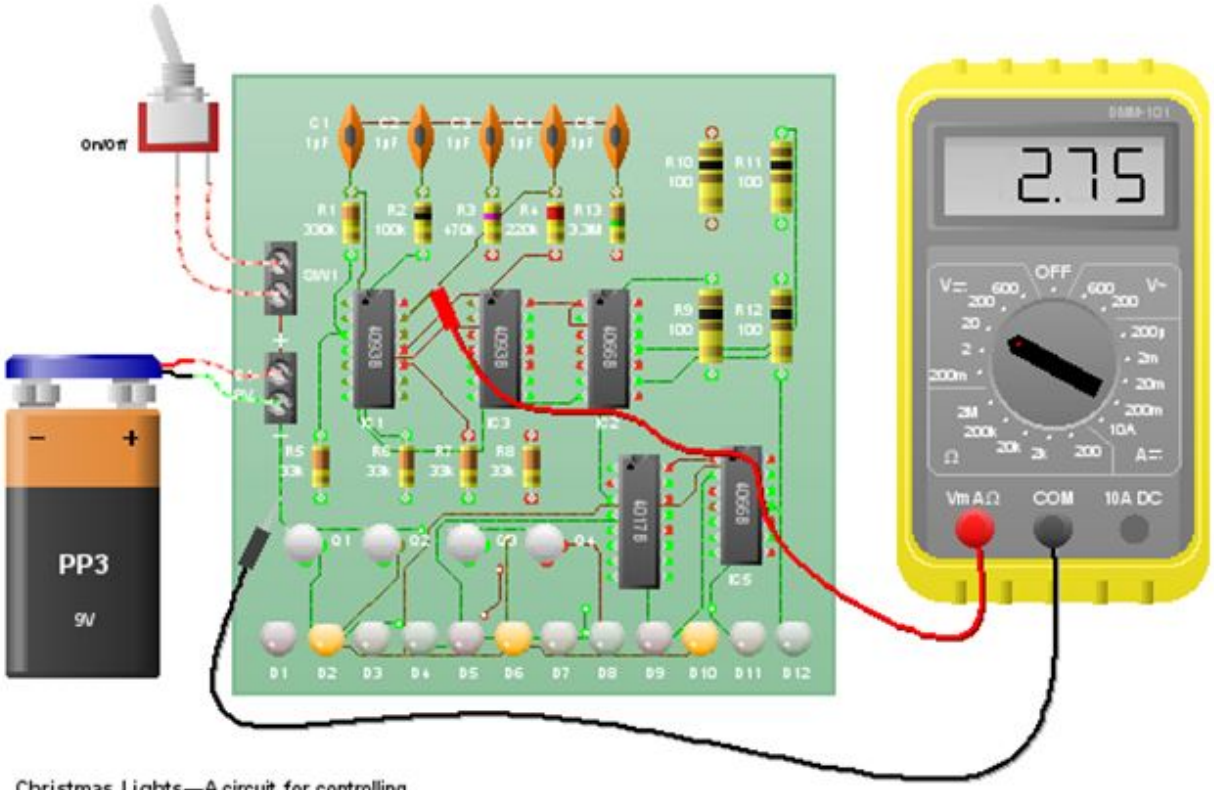
Figura 7: Schema funzionale del MAX1561.

Bibliografia e riferimenti

- Maxim Integrated Products, application note 3256: http://www.maxim-ic.com/an3256
- Maxim Integrated Products, MAX1561: http://www.maxim-ic.com/datasheet/index.mvp/id/3719
- http://it.wikipedia.org/wiki/LED

Introduzione alle porte logiche di base

By Redazione | Fare Elettronica 346 – Giugno 2014



Christmas Lights—A circuit for controlling Christmas tree lights. It makes use of a series of pulsing circuits to provide a sequential LED display.

To see the circuit diagram for this circuit click on the Circuit Diagram tab at the bottom of this window.

E' ben noto a tutti che un sistema complesso digitale è costituito da una combinazione di porte logiche tra esse collegate, dette reti, che rispettano una ben determinata legge che esprime come risultato in uscita una combinazione o parte di essa dei segnali di ingresso. Bene un sistema cosi fatto è chiamata rete logica, ed essa è divisa in due sottosistemi, reti combinatorie e reti sequenziali. Più in generale possiamo dire che una rete logica a priscindere dalla tipologia è costituita da porte logiche elementari, caratterizzate da:

- simbolo logico
- un espressione logica
- e la sua tavola della verità (sistema grafico)

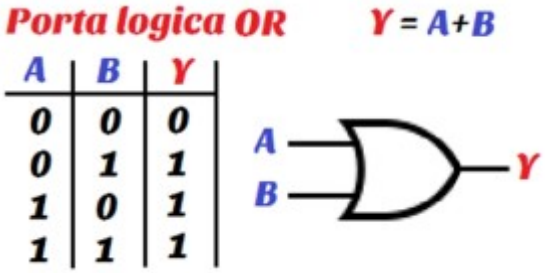
Il simbolo logico viene impiegato negli schemi elettrici delle reti logiche;

l'espressione logica ne esprime il funzionamento con una legge matematica;

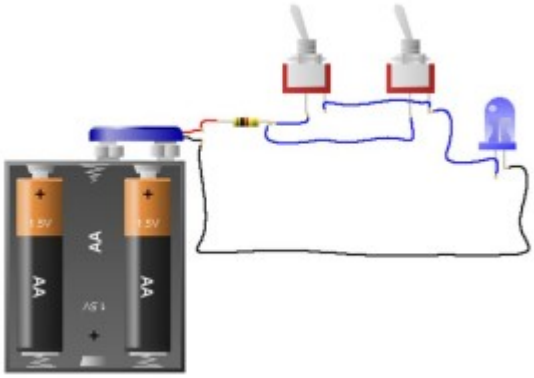
la tavola della verità definisce lo stato dell'uscita in base ai valori assunti dagli ingressi

Un aspetto importante dello studio delle reti logiche è la tavola della verità logica, che come

abbiamo detto definsce lo stato dell'uscita/te in funzione dei valori assunti dalle variabili di ingresso, come visualizzato in questa figura dove è rappresentato il simbolo logico della porta OR e la sua corrispondente tavola della verità:

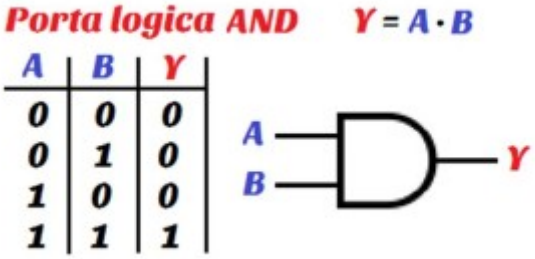


La porta logica OR esegue la somma logica dei valori in ingresso X e Y , si nota subito come l'uscita assume valore logico 1 quando uno dei due ingressi è a valore 1. Per meglio fissare le idee immaginate di realizzare una porta logica OR semplicemente utilizzando una batteria degli interruttori e una lampadina come in figura:

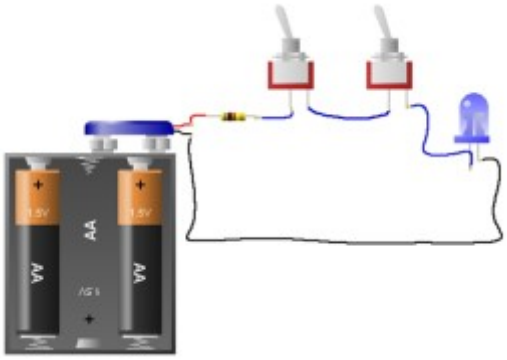


possiamo vedere che gli ingressi sono rappresentati dagli interruttori A e B e l'uscita dalla lampadina. Ora con A e B aperti quindi stato logico zero, la lampadina è spenta, poiché il suo stato è zero, quindi C è zero. Se A o B vengono chiusi, quind si portano allo stato uno allora la lampadina si accende in evidente accordo con il valore logico assunto da C. Questa è una semplice dimostrazione di come funziona la porta logica OR.

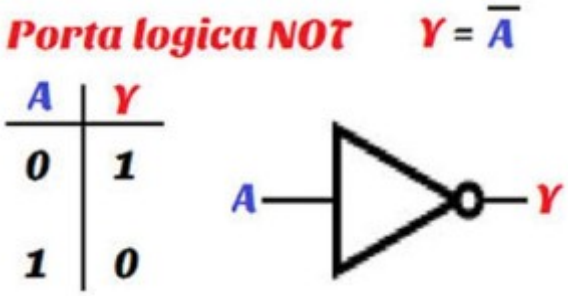
Possiamo fare lo stesso ragionamento considerando la porta logica AND. Questa porta riporta in uscita il prodotto dei valori assunte dalle variabili di ingresso come da figura:



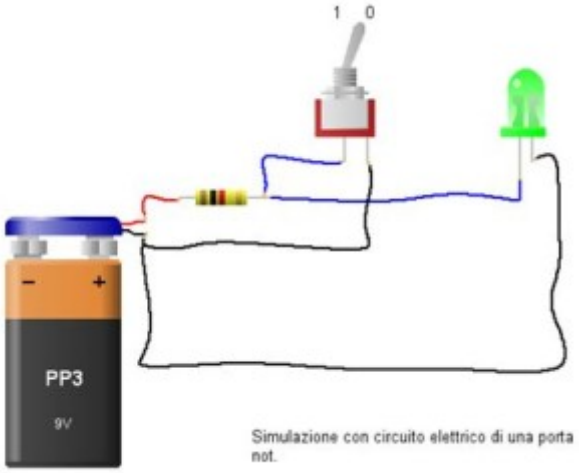
Anche in questo caso possiamo notare come l'uscita assume valore logico uno solo quando entrambi gli ingressi A e B assumono anche essi valore logico uno, esattamente come se fosse un prodotto. Allo stesso modo possiamo rappresentare la porta AND con un circuito elettrico visto in precedenza, ovviamente cambiando il modo di collegare gli interruttori come da figura:



Un'altra porta di base molto importante è la porta NOT, la sua funzione è quella di negazione cioè riporta in uscita la negazione del livello logico di ingresoo, come da tabella:



anche questa situazione può essere rappresentata con l'ausilio di un circuito elettrico per meglio comprenderne il funzionamento.



Come visibile dal circuito abbiamo che se in ingresso ci ritroviamo un livello logico uno, la porta not commuterà in uscita un livello logico zero; similmente nel circuito avremmo lo switch in posizione uno il led sarebbe spento ,quindi idem il viceversa ad un livello logico zero dell'ingresso corrisponde un uscita a livello logico uno o similmente portare lo switch in posizione zero il led sarebbe acceso.

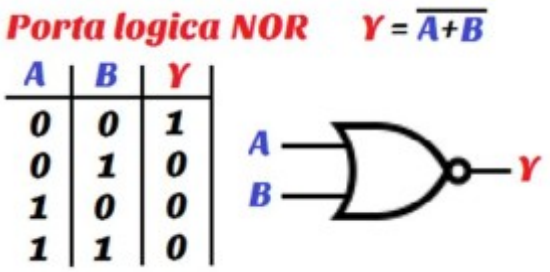
Quanto esposto rappresentano le porte fondamentali o dette anche primitive, sulle quali si sono strutturate le cosidette porte logiche derivate, vediamo in dettaglio quali sono.

INTRODUZIONE ALLE PORTE LOGICHE DERIVATE.

Le porte esposte di seguito sono composte da due o piu' porte fondamentali, pero' per la loro importanza vengono rappresentate con un simbolo proprio. Ricordiamo che una porta logica è

costituita da un simbolo, da una tabella della verità e dalla sua funzione canonica che ne descrive il comportamento. Detto questo possiamo definire la porta logica NOR, come quella porta che ha come funzione la somma negata, o equivalentemente: la somma logica negata si effettua su due o più variabili, l'uscita assume lo stato logico zero se almeno una variabile di ingresso è allo stato logico 1uno In tutti gli altri casi Y=1. Corrisponde ad una porta OR con in cascata una porta NOT.

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta NOR a due ingressi. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta NOR.

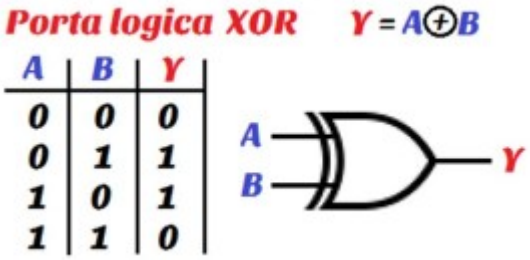


Anche in questo caso possiamo ritrovare senza troppe difficoltà il circuito elettrico di una NOR.

Lascio al lettore la verifica della veridicità dello schema elettrico di una NOR.

La prossima porta derivata è la porta NAND, la sua funzione è di effettuare la negazione del prodotto logico , o equivalentemente il prodotto logico negato NAND si effettua su due o più variabili, l'uscita assume lo stato logico zero se tutte le variabili di ingresso sono allo stato logico uno, In tutti gli altri casi Y=1. Corrisponde ad una AND con in cascata una NOT .

Per finire abbiamo la porta XOR, a differenza delle precedenti porte logiche, l'XOR opera solo su due ingressi. L'uscita vale uno se gli ingressi assumono valore diverso, vale zero se gli ingressi sono tra loro uguali. La funzione logica è rappresentata in figura:

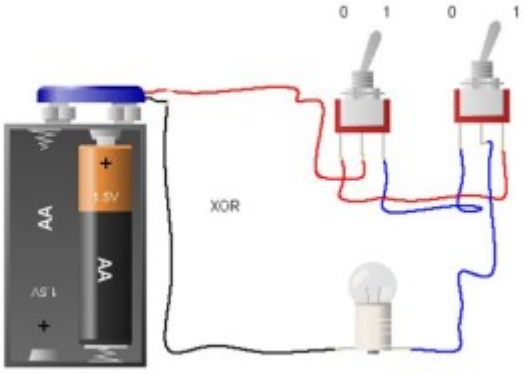


e si legge A or esclusivo B oppure A diverso da B.

Quando si studieranno i modi per ricavare la funzione logica di un circuito digitale a partire dalla sua tabella di verita', si verifichera' che la funzione XOR corrisponde alla funzione:

Y = A B + A B(negato)

Una rappresentazione circuitale della XOR è come in figura:



LE RETI LOGICHE COME MODELLO ASTRATTO DI SISTEMI FISICI

Possiamo definire cosi una rete logica:

Una rete logica `e una macchina a stati finiti, nella quale i simboli di ingresso, uscita e stato sono espressi in notazione binaria.

Informazione è ciò che si conosce di una determinata realtà; per esprimere questa conoscenza occorre un criterio (codice) di rappresentazione, che sia correlato in maniera adeguata alla natura e alle proprietà della realtà in esame. L'elaborazione può essere compiuta da sistemi naturali, primo tra tutti la mente umana, oppure da sistemi artificiali, dei quali il più noto e complesso è il calcolatore.

I sistemi artificiali, a seconda della capacità di elaborazione che devono possedere, possono avere complessità strutturali e funzionali molto differenti, ma tutti sono riconducibili ad un unico modello denominato rete logica.

Una rete logica può essere schematizzata esternamente come una scatola (in inglese black box) con n punti di ingresso ed m punti di uscita, attraverso i quali la rete scambia informazione con l'ambiente: l'informazione viene introdotta nella rete applicando ai punti di entrata n segnali di ingresso e viene fatta uscire come risultato dell'elaborazione prelevando dai punti di uscita m segnali di uscita, proprio come le tabelle delle verità descritte precedentemente. Poiché con la tecnologia attuale le reti logiche sono realizzate convenientemente sotto forma di circuiti elettronici, tutti i segnali in gioco sono di natura elettrica e magnetica e si fa in modo che possano assumere solo valori discreti in numero finito e quindi possano codificare solo l'informazione istanziabile dall'insieme finito di combinazioni di quei valori.

L'algebra della commutazione definisce due postulati fondamentali: il postulato dell'operatore di moltiplicazione logica e quello dell'operatore di addizione logica. La loro descrizione funzionale è fatta mediante tavole di verità, tabelle nelle quali ad ogni

combinazione dei valori delle variabili-operando sono associati i valori delle variabili generate dall'applicazione degli operatori. Ricordando che una sequenza di cifre binarie costituisce la rappresentazione biunivoca in base 2 di un numero naturale, le combinazioni dei valori binari degli operandi possono essere interpretate come le rappresentazioni binarie degli interi nell'intervallo (0, 2^n), essendo n il

numero degli operandi, e quindi una tabella di verità può essere organizzata ordinando tali combinazioni secondo la successione naturale. In sintesi possiamo distinguere nelle reti logiche due sotto sistemi e precisamente:

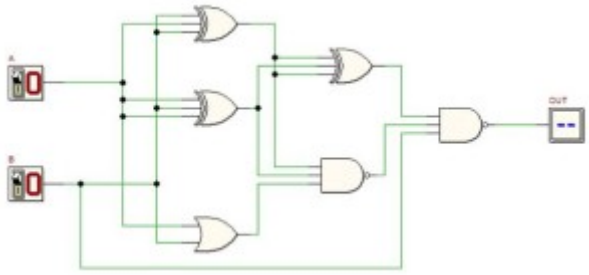
- Reti Combinatorie
- Reti sequenziali

Un calcolatore è costituito da circuiti digitali (hardware) che provvedono a realizzare fisicamente il calcolo. Tali circuiti digitali possono essere classificati in due classi dette reti combinatorie e reti sequenziali.

LE LOGICHE COMBINATORIE

Le reti combinatorie rappresentano l'implementazione di funzioni 'pure' (ossia senza stato) a livello hardware. In particolare, una rete combinatoria implementa una rete logica con n ingressi binari ed m uscite binarie. Poiché una rete combinatoria implementa una funzione, ad ogni combinazione di valori di ingresso corrisponde una ed una sola combinazione di valori di uscita. In un circuito digitale, i valori binari (vero e falso) sono ottenuti tramite discretizzazione dei segnali. In particolare, nella maggior parte dei calcolatori il valore falso (0) è rappresentato dai segnali con voltaggio nell'intervallo [0,1], mentre i segnali con voltaggio in [2,5] rappresentano il valore di verità vero (1). Per facilitare la costruzione di reti combinatorie complesse, tuttavia, è preferibile comporre reti combinatorie predefinite piuttosto che partire dalle porte logiche.

Un esempio di rete combinatoria potrebbe essere il circuito in figura,:

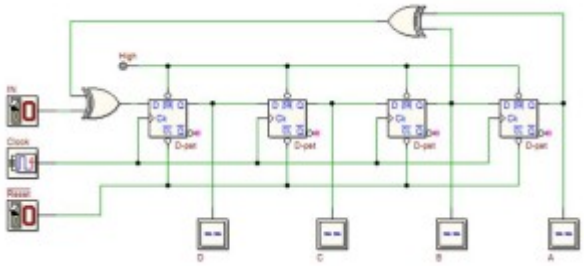


LE RETI LOGICHE SEQUENZIALI

Le reti sequenziali sono reti logiche nelle quali il valore dell'uscita Y ad un certo istante t è funzione sia del valore assunto dalle variabili di entrata in quello stesso istante, sia dei valori assunti dalle medesime variabili prima di t. Con t si sottointende il tempo espresso in us (micro secondi) o ns (nano secondi), ms (milli secondi) e cosi via, istanti prima e dopo aver applicato i livelli logici in ingresso.

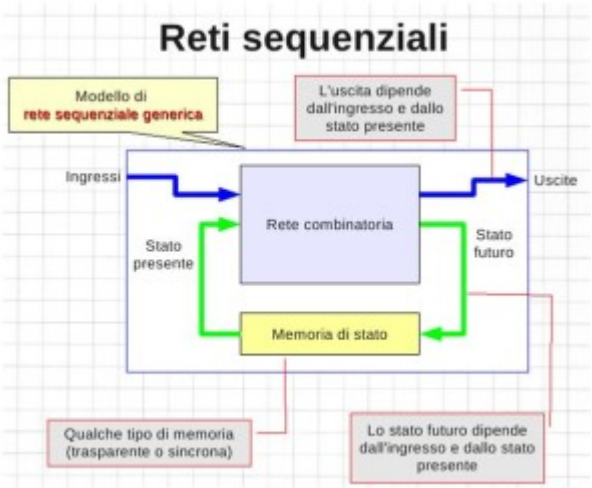
Queste reti hanno quindi memoria

Un esempio di rete combinatoria potrebbe essere il circuito in figura:



Un problema frequente nella risoluzioni o nella determinazione di una rete sequenziale è la sintesi, poiché è strettamente legata agli stati presenti nel ciclo temporale e non sempre di facile soluzione. Ma questo aspetto per ora esula da questa prima trattazione, l'affronteremo nelle sezioni riguardanti l'uso delle FPGA e CPLD.

Ora basti pensare alla rete sequenziale come ad un complesso sistema su cui vige la temporalità, cosi fatto :



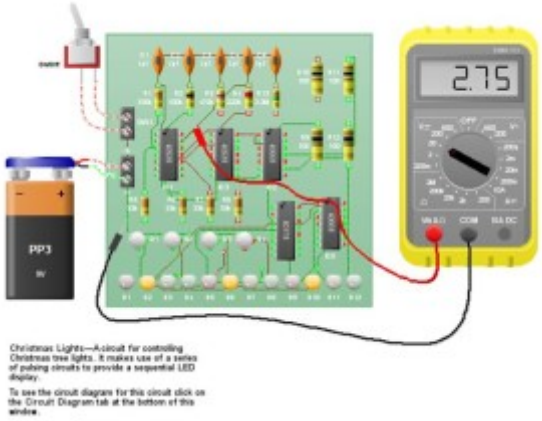
è evidente come l'intero sistema includa inevitabilmente la rete combinatoria, in effetti la rete sequenziale altro non è che una rete combinatoria con memoria temporale (per usare un linguaggio semplice).

ELEMENTI DI LABORATORIO

Come in ogni attività progettuale di fondamentale importanza è il metodo con cui ci si approccia alla progettazione in generale, ma in particolar modo a quella digitale dove i fattori di cui bisogna tener conto sono molti. Pregherei i lettori di provare a progettare reti combinatorie e sequenziali complete di tavola della verità ricorrendo ai software Open Source disponibili in rete come da elenco:

- Qucs
- Logisim
- MultimediaLogic (Windows)
- circuit Wizard (versione free)

questo è il risultato di una rete sequenziale.



NOTE AGGIUNTIVE

Per qualunque problema o richiesta specifica contattatemi pure attraverso la rivista Fare Elettronica, cercherò di soddisfare il più possibile e vostre domande.

Microchip rilascia l’ RN4020 Bluetooth Smart Module

By Redazione | Fare Elettronica 346 – Giugno 2014

Microchip annuncia il suo primo Bluetooth® 4.1 Low Energy module. L’ RN4020 si basa sulla profonda esperienza Microchip riguardo il *Bluetooth Classic* e contiene le regolamentazioni di certificazione internazionali ed è certificato *Bluetooth Special Interest Group* (SIG). Lo stack *Bluetooth Low Energy* (BTLE) integrato ed il supporto on-board per i più diffusi profili *SIG low-energy* accorciano il *time to market* e al contempo assicurando compatibilità Bluetooth, eliminando gli elevati costi di certificazione e riducendo i rischi dello sviluppo. Sul modulo è anche precaricato il *Low-energy Data Profile* (MLDP) di Microchip, che consente ai progettisti di far fluire qualsiasi tipo di dati attraverso il link BTLE.

Poichè l’ RN4020 è un modulo *stack-on-board*, può connettersi a qualsiasi microcontroller con una interfaccia UART, inclusi centinaia di microcontroller PIC®, oppure funzionare in standalone senza un microcontroller, per la raccolta e lo scambio di dati di base, come per i radiofari o sensori. Questo funzionamento standalone viene facilitato dall’esclusivo *no-compile scripting* di Microchip, che consente la configurazione del modulo attraverso una semplice interfaccia di comando ASCII, senza la necessità di alcuno strumento o compilazione.

I progettisti di applicazioni embedded *cost-sensitive* ricercano soluzioni chiavi in mano che rendano facile l’aggiunta di un basso consumo di potenza e di semplicità della connettività Bluetooth LE, che

consente il funzionamento per diversi anni da una sola batteria ed ha una vasta base installata e conforme di smartphone, tablet e computer. Esempi di mercati che richiedono queste soluzioni wireless *low-power command-and-control* includono domotica ed elettrodomestici, medicali e dispositivi indossabili, giocattoli, *tags*, *fobs* e telecomandi, sistemi basati su sensori di pulsazione e prossimità, e persino applicazioni industriali.

Lo Smart Module RN4020 Bluetooth LE include tutto l'hardware, software e certificazioni di cui i progettisti hanno bisogno per aggiungere facilmente connettività *low-energy* a qualsiasi progetto, e al contempo facilitando *End Product Listing* (EPL) attraverso test di compatibilità QDID Bluetooth. Tutti i profili programmabili sono salvati sul modulo e selezionabili liberamente, compresi il flessibile MLD di Microchip e i diffusi profili *Bluetooth SIG low-energy*. In aggiunta ai diffusi profili pubblici, possono essere creati servizi privati attraverso l'interfaccia comandi ASCII. L' RN4020 fornisce anche una antenna integrata nel PCB con 7 dBm di potenza di trasmissione ed una sensibilità di ricezione di -92.5 dBm, che consente una distanza di copertura di 100 metri* in un compatto fattore forma di soli 11,5 x 19,5 x 2,5 millimetri.

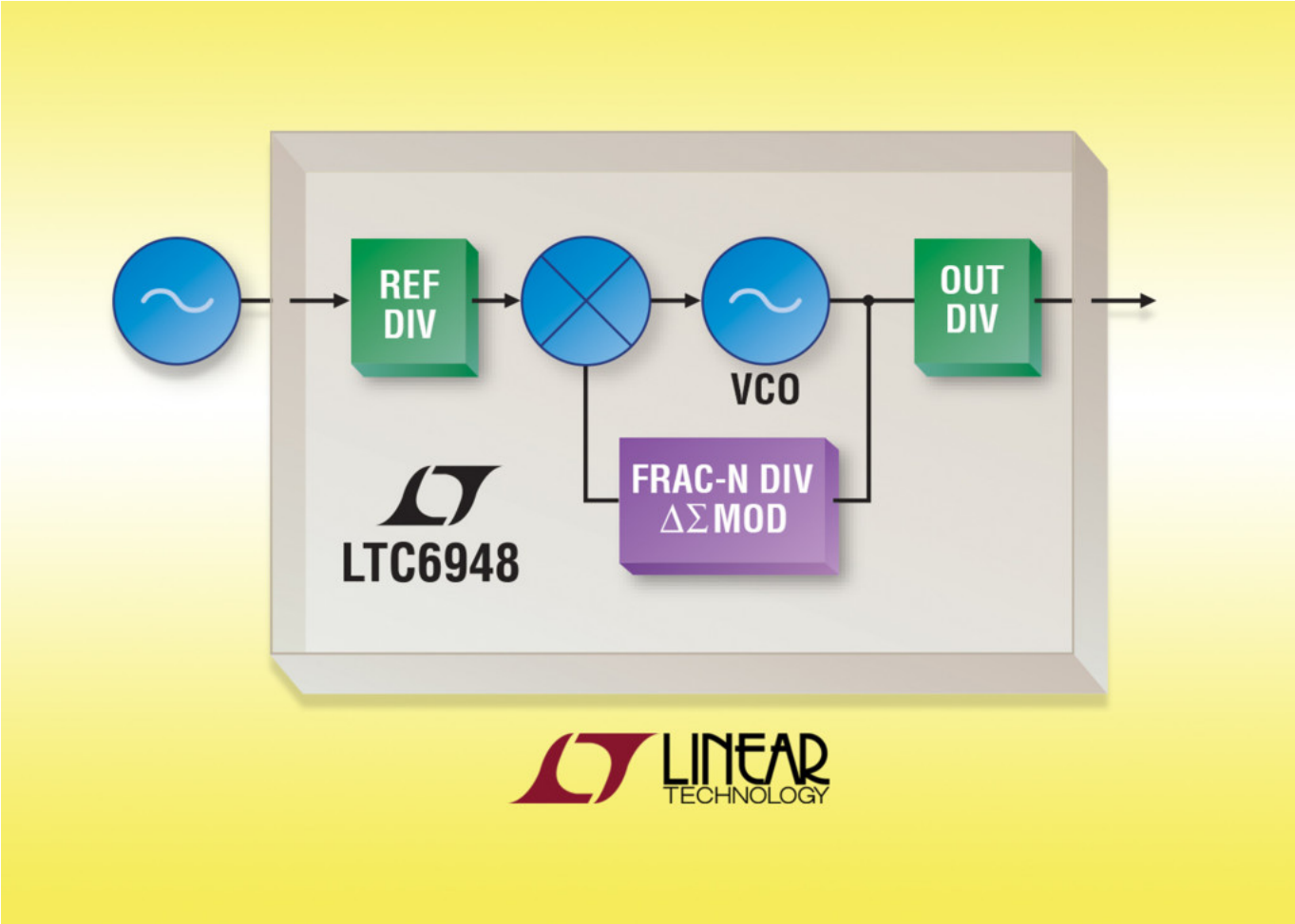
Microchip ha annunciato anche una scheda di supporto per lo sviluppo, la flessibile *RN4020 Bluetooth Low Energy PICtail™/PICtail Plus daughter board* (RN-4020-PICtail) che è disponibile da subito, ad un prezzo di 49,00$. Questa scheda consente lo sviluppo di codice via interfaccia USB verso PC, e sulla interfaccia *In-Circuit Serial Programming™ on-board* per strumenti PICkit™ o MPLAB® REAL ICE™ di Microchip. Oltre a ciò, protegge gli investimenti del cliente in schede di sviluppo Microchip preesistenti con interfaccia PICtail, come la *Explorer 16, PIC18 Explorer* e *PIC32 I/O Expansion Board*. In aggiunta, il microcontroller *eXtreme Low Power* (XLP) PIC18 on-board consente alla scheda PICtail di funzionare in modalità standalone, alimentata via USB dall'host, permettendo dimostrazioni portatili e installazioni di prova.

L' *RN4020 Bluetooth Low Energy Module* è disponibile oggi stesso. Il modulo certificato dagli enti di regolamentazione di tutto il mondo e Bluetooth SIG misura 11,5 x 19,5 x 2,5 millimetri e include una antenna integrata nel PCB.

Maggiori informazioni su: http://www.microchip.com/get/K6QA

Sintetizzatore frazionale N da 6GHz

By Redazione | Fare Elettronica 346 – Giugno 2014



Linear Technology Corporation presenta l'LTC6948, un PLL frazionale N ad elevate prestazioni con un VCO integrato da oltre 6GHz. Al cuore dell'LTC6948 è un modulatore di quarto ordine o delta-sigma che utilizza tecniche intelligenti di modellazione del rumore per ridurre al minimo il contributo di rumore senza creare impulsi di frazionalizzazione nella maggior parte dei PLL frazionali N disponibili sul mercato. Il risultato è un dispositivo che offre tutti i vantaggi della frazionalizzazione – dalla flessibilità della frequenza al rumore di fase complessivo migliorato – ma senza i tradizionali svantaggi associati all'uso di un PLL frazionale N. Inaltri termini, l'LTC6948 è un PLL frazionale N con prestazioni intero N spurie. Se questi vantaggi si abbinano al rumore 1/f e al ridotto rumore di fase inbanda, valori leader del settore, l'LTC6948 risulta ideale per le complesseapplicazioni wireless, di test e misura e militari.

Un buon rumore di fase in banda normalizzato, o cifra di merito, è importante per una soluzione a basso rumore di fase. Tuttavia, il rumore di fase in banda è spesso danneggiato da performance di rumore 1/f del core del PLL. L'LTC6948 vanta una specifica di rumore 1/f normalizzato a 274dBc/Hz senza paragoni che non ostacola lo straordinario valore del rumore in banda normalizzato a -226dBc/Hz. Queste specifiche combinate rendono l'LTC6948 perfetto nelle applicazioni che richiedono un SNR elevato, come le comunicazioni wireless che utilizzano complessi schemi di modulazione, sistemi di comunicazione che utilizzano burst di lunga durata e radar Doppler.

L'LTC6948 riduce il costo totale del sistema, la complessità della progettazione e le dimensioni della

soluzione integrando un VCO con performance competitive del rumore difase e di tutti i componenti PLL, incluso un partitore di riferimento, un rilevatore PDF, una pompa di carica a bassissimo rumore, un partitore frazionale, un partitore e un buffer VCO delle uscite. L'LTC6948 è disponibile in quattro versioni, ciascuna delle quali offre una banda di frequenza VCO diversa e la versione più veloce arriva fino a 6,39GHz. Il partitore VCO delle uscite può essere programmato da 1 a 6 per coprire un'ampia gamma di frequenze fino a 373MHz. La calibrazione VCO viene tipicamente completata poco sopra i 10 µs, un ordine di grandezza più veloce rispetto a molti componenti della concorrenza, cosa che rende l'LTC6948 il PLL d'elezione per le soluzioni che richiedono tempi di assestamento brevissimi.

La progettazione dell'LTC6948 risulta semplificata grazie a FracNWizard™, uno strumento di simulazione e progettazione che può essere scaricato gratuitamente all'indirizzo www.linear.com/FracNWizard. Lo strumento di progettazione FracNWizard aiuta a fornire i valori appropriati dei componenti di filtro del loop semplicemente premendo un tasto e facendo risparmiare ai progettisti ore di calcoli complessi. Prevede in modo preciso le prestazioni del PLL, assistendo il progettista nel processo di debugging e nelle scelte di design.

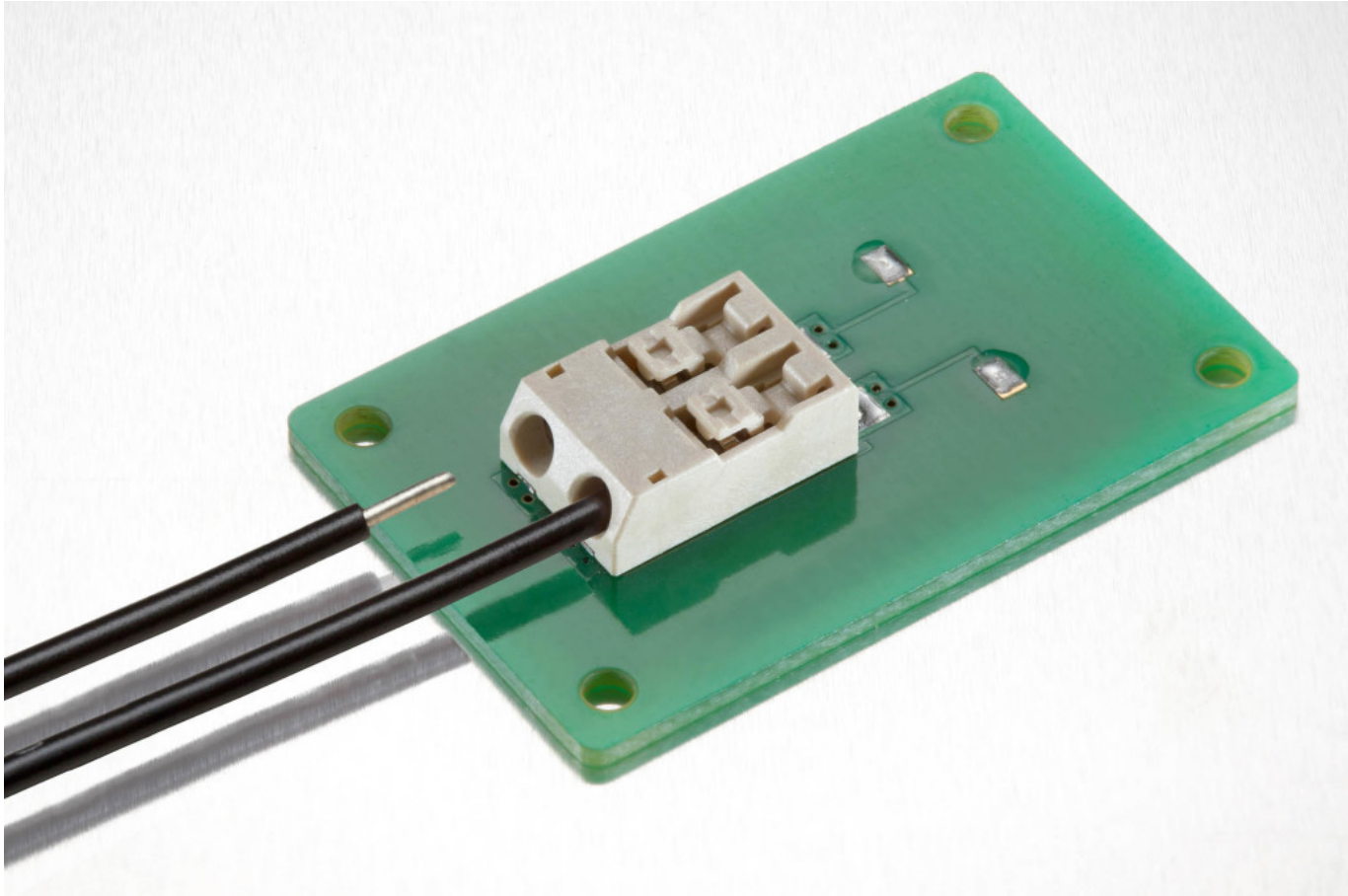
Tutte le versioni dell'LTC6948 supportano temperature operative di giunzione da -40°C a 105°C. I dispositivi sono disponibili in un package QFN in plastica da 4 x 5mm e 28 conduttori. I prezzi dell'LTC6948 partono da $7,25 cad. per 1.000 unità. Il dispositivo è già disponibile a magazzino. Campioni e schede demo sono disponibili su richiesta all'indirizzo www.linear.com/product/LTC6948 o tramite l'ufficio vendite Linear Technology locale.

Riepilogo delle caratteristiche: LTC6948

- PLL frazionale N a basso rumore
- VCO integrato, fino a 6,39GHz
- Denominatore frazionale a 18 bit
- Rumore di fase in banda normalizzato a -226dBc/Hz
- Rumore 1/f in banda normalizzato a -274dBc/Hz
- Rumore di fase in uscita a banda larga di -157dBc/Hz
- Eccellenti prestazioni spurie
- Partitore delle uscite (da 1 a 6, duty cycle al 50%)
- Muting del buffer di uscita
- Corrente della pompa di carica da 1mA a 11,2mA
- Frequenza ingresso di riferimento fino a 425MHz
- Veloce commutazione della frequenza
- Supporto per lo strumento di progettazione software FracNWizard

TTI distribuisce i connettori Molex

By Redazione | Fare Elettronica 346 – Giugno 2014



I connettori Lite-Trap SMT da filo a scheda di Molex per le applicazioni dei moduli d'illuminazione a LED sottili sono ora disponibili tramite TTI, Inc.

Ideali per le applicazioni di moduli d'illuminazione a LED sottili, i connettori da filo a scheda a montaggio superficiale Lite-Trap™ di Molex ora possono essere ordinati presso TTI, Inc., il distributore specializzato leader mondo di componenti passivi, connettori elettromeccanici e discreti. Tali componenti presentano un profilo sottile, permettono la facile rimozione del filo, nonché una forza bassa di inserzione e un'alta forza di ritenzione del filo. Con un profilo di altezza ridotta di 4,20 mm soltanto, i connettori Lite-Trap presentano il profilo più sottile disponibile tra i modelli simili con filo rimuovibile. Altre versioni dal profilo sottile sono disponibili sul mercato, ma questi sono del tipo a connessione permanente irripetibile, che non permettono la rimozione del filo. Il metodo di terminazione del sistema di connettore Lite-Trap è simile ai connettori Wire-Trap di Molex, e anche ad alcune versioni concorrenti del tipo 'Poke-In'. Un filo spelato viene inserito nel connettore e spinge all'apertura un terminale a gabbia che 'intrappola' il filo stesso. Per scollegare, una leva a pulsante sopra l'alloggiamento viene spinto in basso per permettere al filo di disinnestarsi.

Offrendo una forza più bassa d'inserzione dei modelli similari, tali device hanno anche un chiavistello elettronico di uso facilitato che agevola le operazioni per innestare e disinnestare, anche senza l'uso di un attrezzo. Queste caratteristiche sono mirate a offrire un facile assemblaggio sul campo e la rimozione, se richiesto, da operatori addestrati o meno.

NEC LCD Technology nomina Rutronik distributore dell’anno per il 2013

By Redazione | Fare Elettronica 346 – Giugno 2014

NEC LCD Technology (NLT) ha conferito a Rutronik il titolo di “Best Overall LCD Distributor 2013″ per i risultati ottenuti nella distribuzione degli LCD in europa. I display LCD di NLT hanno soddisfatto a pieno gli stringenti requisiti di molteplici applicazioni in campo industriale, medicale, navale a avionico. Il prtfolio NLT include display TFT nelle dimensioni standard da 5.7″ a 15.6″ sia in formato 4:3 che in widescreen. Per i display di NLT sono anche disponibili soluzioni con touch capacitivo. “I nostri prodotti risultano particolarmente adatti ad applicazioni con requisiti molto restrittivi” commenta Thomas Sporkmann Capo del NLT DIsplay Team in Renesas. “I risultati ottenuti in Europa dimostrano che Rutronik è stato un partner molto efficiente che abbiamo deciso di onorare con questo nostro riconoscimento”. Michael Eger, Senior Marketing Manager Displays & Board in Rutronik ringrazia: “I display NLT con il loro eccezionale contrasto, le loro caratteristiche, il loro ampio angolo di visione ed il lungo tempi di vita dell’apparato di retroilluminazione, costituiscono la migliore soluzione per applicazioni in campo industriale, medicale e navale, i principali settori di mercato in cui è focalizzato il nostro business.